

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NÁSTROJE PRO TVORBU PREZENTACÍ

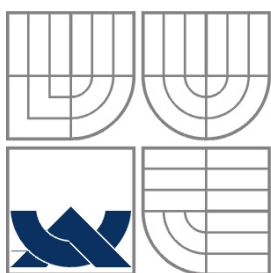
DIPLOMOVÁ PRÁCE

MASTER'S THESIS

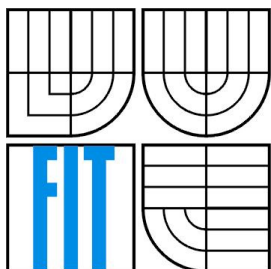
AUTOR PRÁCE
AUTHOR

BC. MARTIN PTAŠEK

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NÁSTROJE PRO TVORBU PREZENTACÍ

TOOLS FOR PRESENTATIONS CREATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

BC. MARTIN PTAŠEK

VEDOUCÍ PRÁCE

SUPERVISOR

DOC. DR. ING. PAVEL ZEMČÍK

BRNO 2010

Abstrakt

Tato práce se zabývá prostředky pro tvorbu pokročilé prezentace. Pokročilými prvky jsou virtuální prostorová scéna, prostorové objekty a video. Shrnuje a popisuje vybrané technologie a možnosti jejich užití. Obsahem práce je také návrh a implementace prezentace pomocí vhodných prostředků. Pro vytvoření je vybrána platforma Flash v operačním systému Windows.

Abstract

This document describes resources for creating an advanced presentation. Advanced parts include virtual three-dimensional scene, three-dimensional objects and video. Describes and summarizes selected technologies and possibilities of using them. Document also includes concept of presentation and its implementation realized by suitable resources. The platform Flash has been chosen for creating in operating system Windows.

Klíčová slova

Prezentace, virtuální scéna, 3D scanner, video, Flash, Silverlight

Keywords

Presentation, virtual scene, 3D scanner, video, Flash, Silverlight

Citace

Ptašek Martin: Nástroje pro tvorbu prezentací, diplomová práce, Brno, FIT VUT v Brně, 2010

Nástroje pro tvorbu prezentací

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Doc. Dr. Ing. Pavla Zemčíka.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Martin Ptašek
25.5.2010

© Martin Ptašek, 2010.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	1
1 Úvod.....	2
2 Současný stav	3
2.1 Teoretický základ	3
2.2 Prostředky vizualizace.....	5
2.3 Silverlight.....	12
2.4 Flash	16
3 Analýza prostředků	22
3.1 Problém kompatibility.....	22
3.2 Použití 3D scanneru	27
3.3 3D v prezentaci.....	35
3.4 Video	38
3.5 Výběr z použitelných technologií.....	45
4 Návrh a implementace	46
4.1 Zásady tvorby prezentace.....	46
4.2 Téma a obsah prezentace.....	48
4.3 Návrh struktury prezentace	48
4.4 Grafický návrh.....	50
4.5 Návrh programové části	52
4.6 Implementace	58
5 Diskuse výsledku	65
6 Závěr	69
Seznam příloh	72
Příloha A.....	73
Příloha B	78

1 Úvod

Současná doba nabízí stále více příležitostí pro rozvoj prezentačních a propagačních technik. Produkty, názory, vše, co má za cíl oslovit větší počet lidí, je třeba vhodně prezentovat. S nárůstem rychlosti komunikace je tato potřeba stále významnější pro zákazníky i firmy, které musejí odolávat konkurenčním tlakům, a prostředkem k získání zákazníků je v tomto případě správná prezentace společnosti či produktu.

Jednou z nejdůležitějších a zároveň nejčastějších forem propagace jsou internetové prezentace. Situace, kdy firma nemá vlastní webové stránky, stejně jako jejich nevhodná forma, design či obsah, může vést ke ztrátě zájmu zákazníků o spolupráci. Tvorba nových prezentací reaguje na jejich stále se zvyšující úroveň vycházející z nároků klientů.

Vývoj probíhá na jedné straně zlepšováním vizuálního stylu, na straně druhé pak využitím nových technologií. Právě technologické aspekty tohoto vývoje jsou hlavním tématem této práce. Podnětem k vytvoření zprávy je celkově nepřilíš přehledný stav zpracování dané problematiky a absence komplexních řešení tvorby prezentací. Výrazná část v současnosti používaných prostředků je určena k produkci prezentací s pouze omezeným využitím modernějších technologií, bez většího uplatnění multimediálního přístupu.

Součástí této zprávy je rozbor stávajícího stavu počítačových technologií zaměřujících se na tvorbu prezentací využívajících pokročilé prvky. Hlavní zaměření se týká problematiky tvorby jednotlivých součástí obsahu prezentace a jejich manipulace za účelem vzniku komplexnější aplikace, která sdružuje větší počet multimediálních prvků s důrazem na uplatnění videa, interaktivních virtuálních scén a prostorových modelů.

Cílem realizace je vytvoření prezentace na vhodné téma demonstrující možnosti vybraných technologických prostředků a jejich účelné propojení.

Úvodní kapitola je nástinem motivů pro řešení dané problematiky, obsahuje také vymezení hlavního cíle a užšího zaměření práce. Obsahem druhé kapitoly je rozbor části prostředků, které je možné v době vzniku této zprávy použít - jedná se o část vycházející ze zpracování semestrálního projektu. Třetí kapitola se zabývá podrobnější analýzou vybraných prostředků v ohledu na jejich použití a případné související problémy. Návrh tématu a struktury výsledné aplikace je přebrán z příslušné části semestrálního projektu, dále se však čtvrtá kapitola věnuje detailnímu návrhu implementace a následně implementaci samotné. Pátá kapitola je zamyšlením autora nad výsledky práce odkazující také na grafický výstup, který je součástí obrazové přílohy. Závěr kompletní zprávy obsahuje shrnutí všech provedených prací a nástin možností, které použití prezentací s pokročilými prvky, případně vytvořené moduly, do budoucna nabízí.

2 Současný stav

Současný stav technologií v oboru vytváření prezentací je shrnut v této kapitole, která je vstupním bodem zkoumání vybraných prostředků vázících se k problematice zpracované v této práci. Vhodné technologie, zběžně popsane v této části, budou hlouběji analyzovány v kapitole 3.

2.1 Teoretický základ

Problematika prezentací s pokročilými prvky je postavena z části na základech jednodušších prezentací, velkou měrou se však opírá o problematiku prostorového zobrazení a práce s videem. V tomto směru se spíše uplatňují poznatky oboru počítačové grafiky, multimédií a uživatelských rozhraní.

Téma této práce se oborově vyskytuje na pomezí všech dříve jmenovaných oborů a kapitola Teoretický základ představuje pouze zlomek výkladu, který by byl nutný pro dokonalý náhled do celkové problematiky. Následující pojmy jsou ve zprávě užívány hojně, případně je vhodné pro tuto práci vymezit jejich užší význam.

2.1.1 Prezentace a prezentační program

Prezentace má v dnešní době široké uplatnění a jako výraz se užívá v mnoha významech. V této zprávě bude výrazem prezentace myšlen audiovizuální výstup, který je používán pro účely propagace, výuky nebo přiblížení nějakého konkrétního problému či myšlenky. Navíc by měla prezentace splňovat určité vlastnosti jako názornost, výstižnost, vhodný vizuální styl a další, kterým se tato zpráva bude zběžně věnovat v kapitole 4 při návrhu výsledné prezentace.

Prezentaci je však potřeba vhodně předvést. K tomu účelu slouží softwarové vybavení, které prezentaci provádí, označované jako prezentační program. Na trhu je množství takovýchto prezentačních programů, přičemž jen některé z nich podporují pokročilé prvky prezentací jako je přehrávání animací, videa, zvuku či náhled 3D objektů, tedy prvky, na které se tato práce zaměřuje především.

Právě internet je tomto směru nejrozšířenějším souborem prezentací podporujících a propagujících pokročilou prezentaci obsahu. Internetové prohlížeče tak tvoří nejpoužívanější prezentační programy vůbec. Navíc jsou uzpůsobeny tak, že je možné v nich s výhodou využít pokročilých prvků prezentace, a to i bez nutnosti připojení k internetu, viz. následující kapitola RIA.

2.1.2 RIA - Rich Internet Applications

Pojem RIA (Rich internet applications) [1] je znám a používán od devadesátých let dvacátého století, kdy byla ve velkém množství vytvářena a zaváděna nová rozšíření pro tvorbu a prezentaci vyspělých webových aplikací.

RIA aplikace, které kombinují širokou dostupnost webu s uživatelským komfortem desktopových aplikací, se od doby vzniku výrazně posunuly kupředu. Je to patrné z dnešní podoby webu i z faktu, že o přízeň vývojářů těchto aplikací se dnes zajímají společnosti jako Google, Adobe, Microsoft nebo Sun.

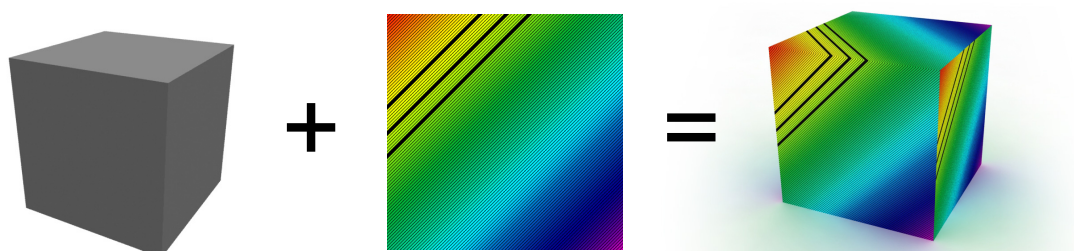
Technologie původně vyvinuté pro internetové prezentace je možné dnes s výhodou použít i jako desktopové nástroje, což přináší pozitiva v použitelnosti i rozšiřitelnosti produktu, kterým je prezentace. Další nespornou výhodou je, že je-li potřebná technologie podporována webovým prohlížečem, je i dobře přenosná mezi různými operačními systémy, neboť stojí na nezávislých zásuvných modulech.

2.1.3 Textura

Texturou může být obecně myšleno vnitřní uspořádání, případně struktura či stavba. V oblasti počítačové grafiky, stejně jako v této práci, je tento výraz používán ve významu vyjadřujícím informaci o vlastnosti povrchu určitého virtuálního tělesa.

Počítačová grafika rozlišuje více druhů textur, které popisují různé vlastnosti jako je barva a intenzita odraženého světla, směr odraženého světla, výška povrchu, průhlednost aj. V této práci se však bude jednat, pokud nebude uvedeno jinak, o barvu povrchu.

Textura je zde chápána jako rastrový obraz, který je na povrch vhodně nanesen tak, že výsledný objekt vypadá, jako by byl tvořen daným materiálem.



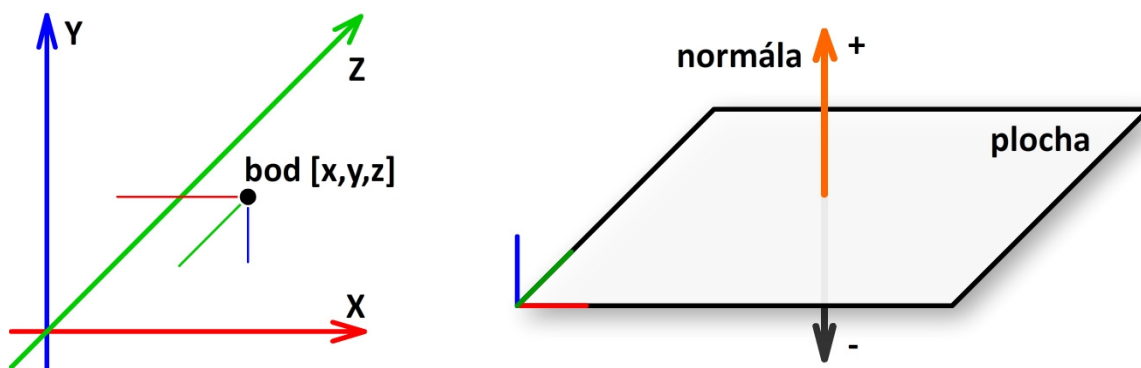
Obrázek 2.1: Ukázka nanesení textury na těleso

2.1.4 3D prostor

V této práci se pojem prostoru objevuje velice často. V případě 3D prostoru se vždy jedná o trojrozměrný prostor, který je modelován v počítači. Virtuální prostor je následně 3D prostor rozšířený o vlastnost interakce uživatele.

V roce 2009 výrazně stoupla popularita 3D projekce, tedy zobrazování 3D prostoru (3D kina, televizory apod.), při kterém jde v největší míře o podání prostorového vjemu divákovi. Jedná se však především o technologii vytváření optické iluze na základě prostorových informací. Pokud se v této práci mluví o 3D prostoru, je pravděpodobné, že v takovém případě máme k dispozici prostorovou informaci a patřičným výpočtem je pak možné vytvářet 3D projekce, ovšem tato technologie je za hranicí tématu této zprávy.

3D prostor v počítači je definován v souřadném systému tří os X, Y, Z. Každý bod musí mít souřadnici $[x, y, z]$. Nejjednodušší plocha je určena minimálně třemi body, navíc většinou obsahuje normálový vektor (normálu), který je kolmý k rovině plochy, určující orientaci plochy. Prostorové objekty jsou ve většině případů (nejde-li o pouhé body) tvořeny plochami, které vytváří jejich povrchy. V oblasti prezentací je možné objem těles (vnitřní strukturu) zanedbat, až na speciální případy, které se zabývají právě vnitřní strukturou. U prezentovaných předmětů je kladen důraz především na povrch (tvar) a texturu (barvu), které by měly mít dostatečnou vypovídající hodnotu.



Obrázek 2.2: Naznačení os souřadného systému (vlevo), plocha s normálovým vektorem (vpravo)

2.2 Prostředky vizualizace

Kapitola se zabývá prostředky vizualizace, tedy technologiemi zobrazování skutečnosti, jejichž výsledky jsou vnímány prostřednictvím zrakových receptorů. Výstupy úzce souvisí s uplatněním zásady názornosti.

2.2.1 3D scanner

3D scanner je moderní prostředek, který umožňuje vytvořit digitální 3D model fyzicky existujícího předmětu, produktu nebo např. živé tkáně. Jde o velice rychlý proces v porovnání s neautomatickým osobním vytvářením virtuálních modelů či jinými technikami, které vedou k téměř shodnému výsledku. Při realizaci výsledné prezentace je využito 3D scanneru Konica Minolta Vivid 910. Jedná se o bezkontaktní 3D snímací zařízení určené pro výzkum, vývoj, oblast designu, školství, video produkci, měření apod.



Obrázek 2.3: 3D scanner Konica Minolta Vivid 910

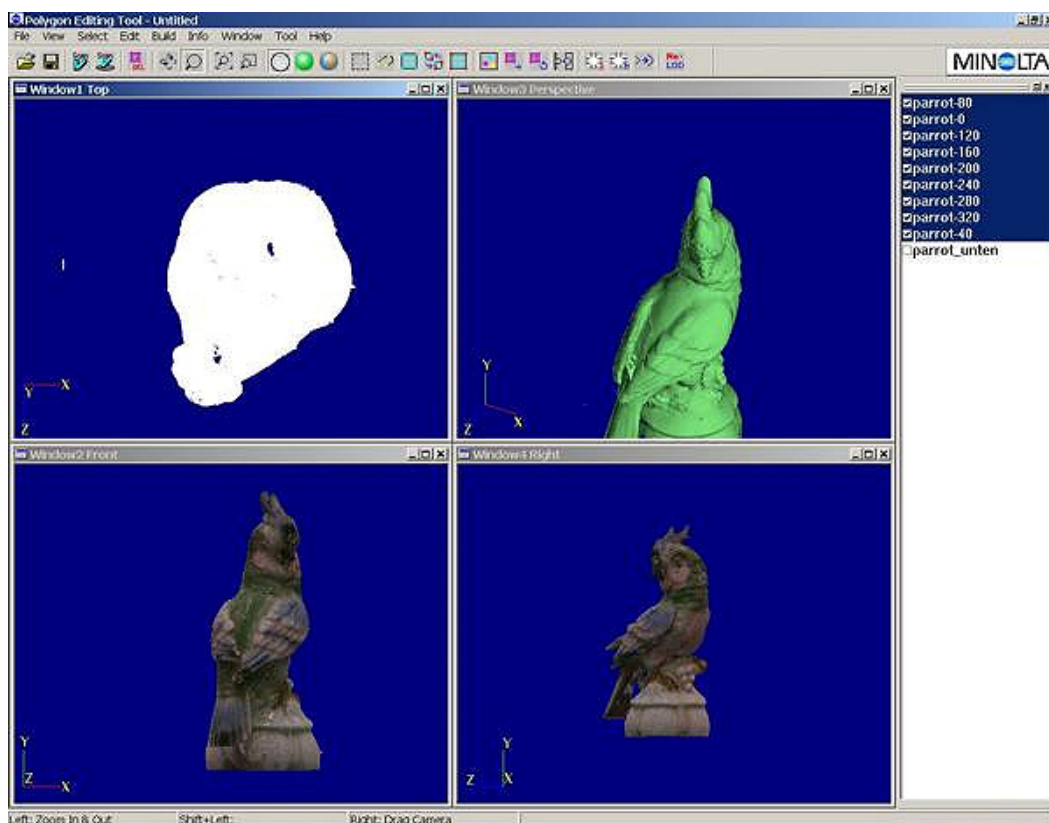
Toto zařízení je určeno pro snímání malých a středně velkých objektů bez lesklých povrchů. Tyto vlastnosti vychází z velikosti zařízení, jeho výkonu a použité technologie měření laserem, který se může od lesklých ploch odrážet a způsobovat tak nepřesnosti měření a výsledných modelů.

Scanner pracuje v prostoru 0,6 až 2,5 metru od samotného zařízení, přičemž objem použitelné oblasti je rozdílný podle použitého objektivu, což znázorňuje následující tabulka:

	f = 25mm	f = 14mm	f = 8mm
osa X	111 - 463 mm	198 - 823 mm	359 - 1196 mm
osa Y	83 - 347 mm	148 - 618 mm	269 - 897 mm
osa Z	40 - 500 mm	70 - 800 mm	110 - 750 mm

Hodnoty jsou limitující pro velikost možného snímaného objektu. Dalším limitním faktorem pro snímání objektu může být čas potřebný pro snímání, protože např. u živého člověka, ať už jde o obličej či jinou část těla, nemůžeme počítat s udržení v nehybném stavu po delší dobu. Doba pořízení jednoho snímku se v rychlejším (méně přesném) režimu pohybuje okolo hodnoty 0,3 sec., což je pro takové měření postačující. Pro úplnost, doba snímání v plné kvalitě je přibližně 2,5 sec. a v barevném režimu (snímá se i barevná textura objektu) je to 0,5 sec.

Přesnost stejně jako chyba měření se pohybuje od hodnoty $\pm 0,1$ mm do hodnoty $\pm 0,22$ mm ve všech třech souřadných osách. Velikost (rozlišení) 3D dat je 307 000 bodů, pro snímání barevného obrazu je možnost pořizovat výstup o rozlišení 640 x 480 pixelů na 24 bitů barevné hloubky.



Obrázek 2.4: Vzhled aplikace pro obsluhu výstupu 3D scanneru

Součástí programového vybavení scanneru je aplikace pro nastavení vlastností snímání a výstupních dat jako rychlost, přesnost nebo např. kroky otáčení objektu před následujícím snímek. Po dokončení skenování je možné ještě data upravit nástrojem Polygon Editing Tool, na obrázku 2.4. Ten obsahuje nástroje pro finální úpravy, doladění orientace a spojení jednotlivých snímků apod. Výstup 3D modelu je pak možný jako přímý otisk naskenovaných snímků, tedy obraz textury a hloubkové mapy. Podporovány jsou i standardní formáty pro přenos 3D dat jako STL, DXF, OBJ nebo VRML.

2.2.2 VRML

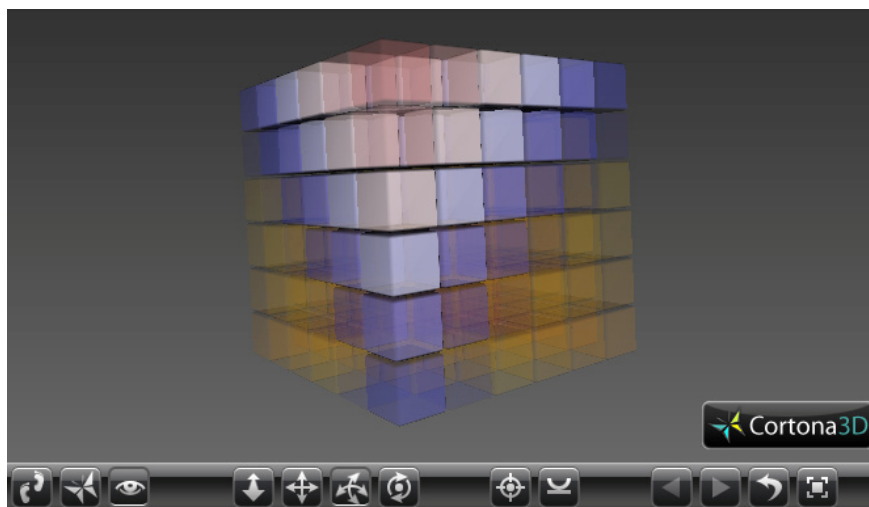
VRML (Virtual Reality Modeling Language) [2] je formátem určeným pro ukládání 3D dat. Je založený na deklarativním programovacím jazyce. Jeho síla spočívá především v možnosti ukládat pasivní prvky i objekty aktivní, ale také možnost jeho použití přímo v internetových stránkách. V takto vytvořeném 3D světě se můžeme procházet, ovládat určitá zařízení a vyvolávat některé předem připravené události. VRML je textovým formátem, ke kterému existuje alternativa využívající komprimace gzip. Díky komprimaci je výsledná velikost zdrojového kódu výrazně zmenšena a paměťovými nároky na uložení se blíží v některých situacích i binárním formátům. Textový formát využívá přípony *.wrl a komprimovaná verze přípony *.wrz.

První verze formátu VRML byla uvedena v roce 1995. Následující koncept verze 2.0 vytvořily firmy Sony a SGI. V témže roce byl koncept standardizován mezinárodní ISO normou VRML 97.

Od této standardizace zůstal formát prakticky nezměněn a dále se nevyvíjí. V současné době je nahrazován formátem X3D, který je postaven na jazyce XML. Zároveň je ale VRML v dnešní době stále používanějším, rozšířenějším a podporovanějším formátem než X3D.

Tvorba a prohlížení VRML

Jelikož je VRML formát víceméně textovým, je možné pro jeho vytváření využít jakéhokoli textového editoru. Pro snadnější editaci byly vytvořeny také specializované editory jako např. Cosmo Worlds nebo RenderSoft VRML editor. Další možností tvorby je využití některého 3D modelovacího nástroje, kde je možné výsledek exportovat do VRML formátu. Mezi takové nástroje patří z nekomerčních aplikací velmi rozšířený Blender, z komerčních řešení je to např. 3Ds Max, Maya nebo Cinema 4D. Pro prohlížení a použití na internetu, kam byl tento formát původně směřován, jsou určeny specializované zásuvné moduly např. Cosmo Player, Cortona3D, Flux Player apod. Jako desktopové aplikace je také možné použít např. aplikace Instant Player, FreeWrl nebo OpenVRML.



Obrázek 2.5: Uživatelské rozhraní prohlížeče VRML Cortona3D a ukázková scéna ve VRML formátu

Struktura VRML formátu

Základní geometrická tělesa, jako jsou krychle, koule, kužel, textová pole apod., je možné generovat pomocí vyhrazených klíčových slov, které reprezentují tzv. uzly modelu. Použití klíčových slov umožňuje úsporný a přehledný zápis VRML souboru. Není tak nezbytné tyto objekty rozkládat na trojúhelníkové plochy a tento rozklad je zcela ponechán na programu, který daný objekt vykresluje. Je tedy patrné, že rychlost a kvalita zobrazeného objektu hodně závisí na kvalitě použitého prohlížeče. Ostatní prostorová tělesa jsou popsána pomocí seznamu souřadnic vrcholů a ploch, jejichž vrcholy jsou specifikovány indexy do seznamu vrcholů.

Celá virtuální scéna je tvořena hierarchickým grafem scény, který tvoří jednotlivé struktury popisující vlastní obsah.

Formát VRML dovoluje ve scéně definovat jednoduché animace objektů, jako rotace či posun v prostoru, zároveň ale obsahuje i dynamické prvky, které rozšiřují interakci s uživatelem, tzv. senzory. Senzory jsou uzly použité pro podmíněné řízení dějů ve scéně a dělí se na dva typy:

- dragging - generují události při změně pozice senzoru například otočením nebo posunutím
- standard - vyvolávají události za různých jiných okolností, jako je například viditelnost, úroveň jasů okolí, dotyk předmětu... .

Základní senzory se rozdělují následovně:

CylinderSensor	Dragging
PlaneSensor	Dragging
ProximitySensor	Standard
SphereSensor	Dragging
TimeSensor	Standard
TouchSensor	Standard
VisibilitySensor	Standard

Nevýhodou ovšem zůstává, že některé prohlížeče určité typy nepodporují.

K další zajímavé vlastnosti, která přibyla ve verzi 2.0 je možnost reálnějšího průchodu scénou virtuální reality pomocí tzv. avatara. Jde o dynamický objekt kamery, lze si ho představit jako návštěvníka scény, který může mít definovány parametry, jako výška, šířka, rychlost apod. To je výhodné při simulaci průchodu člověka. Nedostatkem ovšem zůstává nemožnost interakce více různých uživatelů v jedné scéně.

2.2.3 X3D

X3D (Extensible 3D) [3] je otevřený formát určený k ukládání objektů 3D scény. Hlavní myšlenkou vzniku tohoto formátu bylo rozšíření formátu VRML, z něhož X3D vychází a řeší některé nedostatky, které VRML má.

X3D stejně jako XML využívá stromovou strukturu pro ukládání objektů scény. Jde o velice jednoduché a přehledné provedení, kde je možné měnit vlastnosti objektů, vytvářet události (provázanost objektů), skriptovat objekty (např. vytvářet animace).

Díky tomu, že je tento projekt otevřený, mají přístup k syntaxi všichni uživatelé, což by mělo pomoci s rozšířením a rychlejším rozvojem než kdyby byl vyvíjen pouze úzkou skupinou programátorů. Na příspěvky pro rozvoj tohoto formátu dohlíží skupina Web3D Consortium.

Nevýhodou tohoto formátu ale stále zůstává slabá podpora ze strany komerčních zákazníků, a tak v současné době neexistuje větší množství prohlížečů, zásuvných modulů do internetových prohlížečů ani editorů tohoto formátu.

2.2.4 Programy pro zachycování videa

V současné době existuje několik softwarových řešení pro zachycování videa. Přesněji, jde o aplikace, které se zaměřují především na snímání obrazu výstupního zařízení (většinou obrazovky počítače), nebo pouze jeho zvolené části. Tyto snímky současně převádí na výsledný video výstup.

Adobe Captivate

Je komerční aplikace firmy Adobe, která původně vycházela právě z projektu CamStudio. Jde o komplexní aplikaci, která je určena nejen pro snímání obrazovky, ale především pro vytváření interaktivních výukových programů. V tomto směru jde o software, který je svým rozsahem výjimečný, ale pro účely samotné prezentace ve smyslu, kterým se zabývá tato práce, jsou některé jeho funkce nadbytečné.

Aplikace Captivate umožňuje veškerá nastavení jako aplikace CamStudio s rozdílem právě v možnostech tvorby výukových prezentací. Mimo snímání videa je zde možnost snímat i interakce ovládacích prvků. Ve výsledku je možné vytvořit prezentaci, která nejen ukazuje funkce předváděného softwaru, ale může dokonce čekat na interakci uživatele, tedy dokud nezareaguje, případně i dokud uživatel neprovede definované kroky.

Možnosti definované interakce k aktivaci a přehrání další části prezentace je několik. Nejběžněji je definováno čekání prezentace dokud uživatel nepřesune kurzor na zvolenou pozici nebo nestiskne určité tlačítko. Veškeré kroky je možné doprovodit zvukovou nahrávkou, kterou je možné pořídit přímo při snímání videa, nebo lze vkládat zvuky externí. Další možností je vkládání komentářů a nákrešů například pro potřeby výuky.

CamStudio

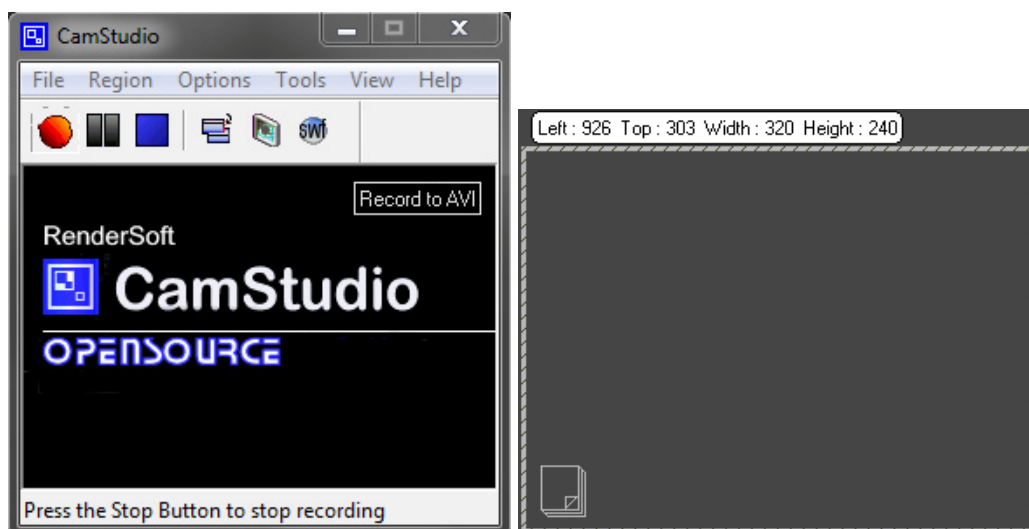
CamStudio [4] je jedna z nekomerčních aplikací pro zachycování videa šířená pod licencí GPL. Aplikace byla původně vydána společností RenderSoft, později ji ale odkoupila společnost Macromedia, která měla zájem o aplikaci RoboDemo využívající právě CamStudio (později pod Adobe přetvořeno na Captivate).

CamStudio je určené k vytváření demonstračních videí jakéhokoli softwaru, k vytváření video návodů nebo výukových ukázek. Uplatnění je možné i při řešení opakujících se problémů s počítačem či dalších možných specifických úlohách.

Tento software dokáže snímat veškerou audio, ale především video činnost na ploše počítače a vytváří výstup ve standardním .avi formátu. Následně také dokáže převádět video formát .avi na .swf, což je formát používaný jako výsledná aplikace flash technologie. Při konverzi je nastavitelná kvalita výsledného videa, kterou je možné značně komprimovat pro potřeby streamování videa na internetu.

Aplikace dovoluje volbu snímacího pole v předvolené velikosti 320 x 240 pixelů, v celé šíři obrazovky podle nastaveného rozlišení nebo ve zcela libovolném rozměru určeném uživatelem. Další nastavení dovolují určit kvalitu snímaného videa a jeho snímkovací frekvenci. Při nastavení velké snímací plochy a vysoké snímací frekvence je na počítači znát značné zatížení, především kvůli potřebě nasnímaná data zpracovat a uložit.

Zajímavou a zároveň užitečnou funkcí je možnost nesnímat kurzor myši. Ten je na obrazovce vidět, ale ve výsledném videu se neobjeví. V opačném případě lze ponechat kurzor viditelný i ve videu, přičemž existuje možnost zapnout i jeho zvýraznění, což je výhodné zejména ve výukových videích.



Obrázek 2.6: Vzhled aplikace CamStudio a volby nahrávacího rámu

2.3 Silverlight

Je poměrně nový produkt mezi aplikacemi RIA (od roku 2007). Jedná se o zásuvný modul vyvíjený společností Microsoft. Je navržen tak, aby dokázal překlenout nedostatky HTML (Hyper Text Markup Language) webových prezentací a poskytl k vývoji prostředek, jehož pomocí je možné vytvořit graficky bohatší interaktivní aplikace. Dovoluje tak kombinovat text, bitmapovou a vektorovou grafiku, animace i video.

2.3.1 Verze Silverlight

Silverlight [5] byl vydán v několika verzích a v současné době je podporován i na mobilních operačních systémech Symbian a Windows Mobile. Původně Silverlight staví na dřívější technologii WPF/e (Windows Presentation Foundation/Everywhere). První verze zahrnovala podporu prohlížečů IE, Firefox a Safari na platformách Windows a Mac. Podporu Linuxových systémů přinesl projekt Mono (Moonlight), který je otevřenou verzí implementace Silverlight.

Další verze Silverlight přináší rozšíření podpory vývojářů a tvůrců obsahu. Je implementována implicitní podpora Linuxových systémů a především rozšířena podpora .NET Framework s Rich Base Class knihovnou společně s jazyky C#, Visual Basic, JavaScript, IronPython, IronRuby a PHP (kompilováno pro .NET kompilátorem Phalanger).

Zatím poslední dokončená verze 3 [6], vydaná 10. 7. 2009, přinesla množství rozšíření:

- Rozšířená datová podpora a nástroje pro práci s daty:
 - Podpora pro business objekty jako je jednoduché stažení, třídění, filtrování a stránkování.
 - Validace vstupu je realizována obdobně jako v ASP.NET. V Silverlight je tak možné validovat vstup uživatele a vizuálně jej informovat o jeho chybách.
 - Provázání vlastností ovládacího prvku s vlastností jiného prvku nebo objektu přímo v XAML kódu.
 - Datový formulář je také novým ovládacím prvkem, který podporuje pole (fields), validaci, updatování a stránkování dat.

- Schopnost běhu mimo prohlížeč

Silverlight dnes nabízí možnost instalace aplikace přímo do lokálního operačního systému. Na rozdíl od běžných desktopových aplikací ale běží v sandboxu, který zamezuje spouštění kódu nedobrovolně staženého z internetu. Lokální instalace se z pohledu koncového uživatele chová jako desktopová. Navíc pracuje bez nutnosti instalace jakéhokoli dalšího softwaru a funguje dokonce i bez dočasného připojení k internetu.

Pro vývojáře jsou připraveny následující technologie:

- Nasazení aplikace jedním klikem s umístěním odkazů do start menu a na plochu.

- Díky sandboxu není nutná změna přístupových práv při instalaci aplikace. Stejně tak nejsou vyžadována dodatečná práva při běhu aplikace.
- Automatické aktualizace aplikace po jejím startu. Je-li připojení k internetu dostupné, je nová verze detekována, stažena a nainstalována.
- Je detekováno síťové připojení a na jeho základě je možné rozhodnout, zda data poslat přímo na server nebo dočasně uložit na disk.
- Vysoká RIA produktivita
 - Všech více než 60 ovládacích prvků bylo uvolněno se svými zdrojovými kódy jako projekt Silverlight Toolkit.
 - Podpora SEO (Search Engine Optimization, optimalizace pro vyhledávače). Vyhledávání na internetu a RIA technologie často nejsou příliš kompatibilní. V Silverlight je nově tato otázka řešitelná pomocí business objektů na straně serveru společně s ASP.NET ovládacími prvky a mapou webových stránek. Díky nim lze datově řízený klientský RIA obsah převést na snadně indexovatelný HTML obsah.
 - Podpora záložek uvnitř aplikace je vylepšením, zejména z pohledu koncového uživatele.
- Nové grafické efekty
 - Podpora hardwarové akcelerace přinesla přenesení výpočetního výkonu z CPU na GPU.
 - Podpora 3D je umožněna tzv. perspektivním 3D, které umožňuje v 3D prostoru pohybovat rovinou, na které jsou umístěny jednotlivé elementy. Vedle posunů, rotací nebo zvětšování jsou dále k dispozici fronty ve 3D a přechody.
 - K dispozici jsou jednoduché shadery jako rozmazání nebo vržení stínů. Jejich architektura je však otevřená a lze napsat a použít vlastní. Shader lze navíc aplikovat nejen na bitmapovou mapu, ale i video nebo ovládací prvky.
 - Přibyl nové bitmapové API. Díky němu je možné manipulovat s jednotlivými pixely bitmapových map. S výhodou lze použít pro snímání videa, datovou vizualizaci nebo algoritmické generování obsahu.
 - Při vytváření tématických vzhledů je možné použít typické kaskádování stylů jako u CSS.
 - Animace v předchozích verzích Silverlight neumožňovaly simulaci přirozeného pohybu, jako jsou například odrazy. Zavedením tzv. Easing funkcí, lze nyní realisticky simulovat tyto pohyby. Dokonce lze pohyb popsat vlastními matematickými funkcemi pro simulaci libovolného chování.
 - K dalším rozšířením patří například rychlejší vykreslování a animace textu, případně možnost využití lokálních fontů.
- Podpora pro video a audio ve vysoké kvalitě

Silverlight byl od svého počátku v oblasti videa silnou technologií, zejména díky VC-1 kodeku a kombinace s Windows Media Services. Ve verzi 3.0 byla přidána velice významná rozšíření.

- Byla přidána podpora MPEG4 ve formátu H.264 s ACC pro vysoce kvalitní audio.
- Podpora GPU akcelerace pro vykreslování HD videa.
- Silverlight nabízí možnost využít své vlastní kodeky nebo kodeky třetích stran, díky otevření AV pipeline. Pomocí Raw AV Pipeline lze dekodování streamu předat externí komponentě a samotné vykreslení opět ponechat na Silverlightu.
- Zvýšená výkonnost

Mimo vlastností, které se vyskytují v předchozím výčtu a mají také podíl na zvýšení výkonnosti aplikací Silverlight je ještě několik dalších, které k výkonnosti také zásadní měrou přispívají.

 - Schopnost lokálně ukládat aplikační knihovny tak, že mohou být sdíleny mezi jednotlivými aplikacemi, aniž by se musely opakovaně stahovat. Knihovny jsou staženy na lokální počítač při prvním přístupu a následně zpřístupněny i ostatním Silverlight aplikacím. Toto nastavení lze aplikovat jak na své vlastní knihovny, tak na originální knihovny Microsoft.
 - Vylepšený Deep Zoom. Tato technologie poprvé uvedená ve verzi 2.0 byla nyní zdokonalena a daleko plynuleji umožňuje vykreslovat obrázky v extrémně vysokém rozlišení.
 - Komunikace prostřednictvím XML má celou řadu výhod, ale i jednu nevýhodu - objem dat. Pokud nám na tomto parametru záleží, což je vždy, když jde o výkon, můžeme s výhodou využít binárního XML. Data jsou při této formě komunikace komprimována a jejich přenos je významně zrychlen.
 - Díky Local Connection není nutné s daty putovat z jedné komponenty na server a zpět do komponenty druhé. Snadno tak může např. předávat data list box v jedné aplikaci grafu nebo datagridu v aplikaci druhé.

Posledním vývojovým krokem bylo vydání beta verze 4 [7], která byla vydána čtyři měsíce po verzi 3.0. Silverlight 4.0 obsahuje nová rozšíření, která mají vývojářům umožnit tvorbu aplikací pro firmy i běžné uživatele. Nabídne zejména rozšířené možnosti běhu mimo prohlížeč bez potřeby změny kódu programu, plnou podporu pro Microsoft Visual Studio 2010, přístup k Microsoft Office a Microsoft SharePoint Serveru, ale také vylepšené možnosti pro tisk, tvorbu sítí, reportů i nejrůznějších schémat. Beta verze Silverlight 4 dále nabízí využití webových kamer a mikrofonů, přináší podporu pro off-line přehrávání chráněného obsahu.

2.3.2 Architektura Silverlight aplikace

Silverlight aplikace se skládá ze tří částí. Jedná se o hostitelskou část, dále soubor obsahující značkovací jazyk XAML a dále soubory aplikační logiky, které mohou být psány v jednom z podporovaných programovacích jazyků.

Hostitelská část

HTML/XHTML stránka je pouze hostitelem Silverlight aplikace. To znamená, že webová stránka neobsahuje kód Silverlight aplikace přímo, místo toho obsahuje základní HTML nebo XHTML značky a kód JavaScriptu, který obsahuje nutné funkce pro zpřístupnění obsahu Silverlightu. Z tohoto důvodu nemůže být obsah Silverlightu zobrazen uživatelům s prohlížeči, které mají vypnutý JavaScript nebo jej nepodporují. Pro následné spuštění Silverlight aplikace z hostitelské stránky je nutné mít v prohlížeči nainstalovaný plugin. Jak již však bylo dříve řečeno, nové verze Silverlight podporují spuštění aplikací i bez potřeby webového prohlížeče.

XAML

Jedná se o značkovací jazyk, používaný pro tvorbu uživatelského prostředí, založený na XML. Tento jazyk byl navržen primárně pro potřeby vývojářů, kterým umožnil sestavit uživatelské rozhraní ve WPF (Windows Presentation Foundation), kde je Silverlight malá podmnožina WPF, využívající stejný standard značek. Konceptně je XAML analogií k HTML.

Aplikační logika

Poslední částí Silverlight aplikace je vnitřní logika, která slouží pro vytvoření obsluhy událostí jednotlivých prvků rozhraní. Tyto prvky jsou definovány pomocí značkovacího jazyka XAML. Spojení obsluhy událostí a XAML je umožněno pomocí atributu třída u zobrazovacího plátna.

Pro psaní aplikační logiky lze ve verzi 1.0 použít pouze JavaScriptové funkce, které jsou následně ze Silverlight aplikace volány. Ve vyšších verzích jsou již možnosti psaní vnitřní logiky výrazně větší. V XAML souboru lze definovat zdroj aplikační logiky. Tímto zdrojem může být kompilovaný kód nebo skript. Z kompilovaných jazyků jsou podporovány C# a VB.NET, jejichž běh zajišťuje CLR (common language runtime) integrovaný v zásuvném modulu. Ze skriptovacích jazyků jsou podporovány JavaScript, Python nebo Ruby.

Grafické možnosti a animace

Prvky grafického uživatelského rozhraní Silverlight použitelné při vývoji aplikací jsou z velké části přebrány WPF, jde o standardní prvky obsažené v .NET Frameworku pro Silverlight.

Základ další funkcionality je tvořen podporou kreslení, což umožňuje vytváření vlastních ovládacích prvků, interaktivní grafiky i animací. Obsaženy jsou základní primitiva jako obdélník, elipsa, linka apod., které mají společnou množinu některých vlastností. Mimo to je možné základní prvky seskupovat ve složitější objekty.

Možnost vytvářet animace významným způsobem rozšiřuje možnosti vytvářet dynamická uživatelská rozhraní. Animace mohou být založeny na čase, tedy výsledná animace je vytvořena jako lineární pohyb mezi definovanými místy za zvolený čas. Dále je možné, podobným způsobem jako čas, určit vlastnost, která se v určitém intervalu mění nebo je možné použít definovaných animačních tříd.

Změny vlastností, jako jsou různé geometrické transformace, je možné provádět pomocí transformačních tříd, jimiž se rozumí objekty, které mění způsob vykreslování tvaru nebo prvku tím, že změní použitý souřadný systém pro vykreslení. Takto je možno prvky otáčet, deformovat, měnit velikost apod. Tyto vlastnosti jsou chápány jako vlastnosti objektu, které je možné taktéž animovat.

2.4 Flash

Adobe Flash (původně Macromedia Flash) [8] je hlavním konkurentem, dříve popsané technologie Silverlight. Jde o multimediální platformu vyvíjenou a distribuovanou společností Adobe Systems, která se od doby vzniku v roce 1996 stala velice úspěšnou, především na internetu. Flash je technologie, která se vyvíjí již delší dobu, s čímž souvisí jeho mnohem výraznější rozšíření a větší povědomí uživatelů než u Silverlight. Původně sloužil pro tvorbu jednoduchých animací založených především na vektorové grafice, přehrávání videa a tvorbu působivější reklamy na internetu jako alternativa k tehdy velice oblíbenému formátu .gif. V dnešní době je Flash jedním z nejsilnějších hráčů na poli RIA aplikací a zásadní měrou rozšiřuje možnosti interaktivity internetu.

Kromě manipulace s vektorovou i rastrovou grafikou je zde možnost obousměrného streamování audia i videa. Již tak široké možnosti manipulace s interaktivním multimediálním obsahem byly v poslední době dále rozšířeny o 3D rozměr a tvorba v tomto směru je podporována mnoha společnostmi, které vyvíjí prostředky pro tvorbu, zobrazování, interakci v prostorovém virtuálním světě. Příkladem mohou být aplikace pro prohlížení 3D prostorů, ať již z reálných míst, nebo vymyšlených, dále hry, které se kvalitou zpracování blíží nedávným desktopovým titulům, s implementací reálného fyzikálního světa.

Podpora technologie se týká všech rozšířenějších platforem (Windows, Mac, Linux), včetně značného rozšíření i na jiná než desktopová zařízení jako jsou mobilní telefony či konzolová zařízení (technologie Flash Lite).

2.4.1 Vývoj Flash

Před samotným vývojem Flash společností Macromedia, byla snaha Jonathana Gaye rozvinout myšlenku intuitivního kreslení na počítačích s operačním systémem PenPoint. Když ale PenPoint na trhu neuspěl byla aplikace SmartSketch portována na Windows a Mac. Zároveň byla rozvíjena myšlenka vektorové grafiky a jejích animací, vhodných svým malým objemem dat pro internet. To dalo v roce 1996 vzniknout aplikaci FutureSplash Animator, kterou ještě téhož roku převzala společnost Macromedia. Následuje kontinuální vývoj editoru Macromedia Flash, společně s přehrávačem Flash Player. Značnou částí vývoje, přibližně do roku 2005, je spojena technologie Flash výhradně přímo s editorem. Po převzetí firmou Adobe a jejím částečném uvolnění specifikací formátu, je však možné vytvářet obsah i v editorech třetích stran, které svůj obsah dokáží kompilovat tak, aby mohl být přehrán přehrávačem Flash.

Původní editor, určený především designérům, bez potřeby hlubší znalosti programování, dal postupem času vzniknout technologii, která se svými možnostmi může rovnat moderním programovacím jazykům a systémům. Stalo se tak vývojem programovacího jazyka ActionScript, který v začátcích, ve verzi 1.0, sloužil především k jednoduchému ovládání prezentací. Verze 2.0 již přinesla objektově orientovaný přístup a značný nárůst výkonu. Zároveň se však snažila zachovat dostatečnou volnost designérům a animátorům, a tak nebyla z pohledu programování dostatečně čistá. Dnešní vývoj a potřeba nárůstu výkonu aplikací, především kvůli tvorbě 3D prostředí, zapříčinily vznik verze 3.0, která je již zcela programátorsky čistá a je možné ji přirovnat k Javě či C#.

V dnešní době jsou světově nejrozšířenější aplikace a komponenty postavené na programovacím jazyku ActionScript 2.0, jsou zde zastoupeny jednodušší aplikace, reklamy či webové stránky. ActionScript 3.0 pak využívají především programátoři k tvorbě her či aplikací spojených s výraznější komunikací se servery apod.

Ve výsledku dokáže Flash řešit shodné problémy jako technologie Silverlight, ovšem technologie Flash přišla o mnoho dříve, takže se předpokládá dokonalejší odladění některých problémů jako komunikace po síti, bezpečnost apod. Navíc je zatím Flash stále více rozšířen a s rychlým přechodem vývojářů Flash k Silverlight se nepočítá, protože dosud Silverlight pouze doháněl Flash, který byl technologicky značně napřed.

2.4.2 Architektura Flash

Flash aplikace je možné používat dvěma způsoby. Nejrozšířenější je spouštění přes internetový prohlížeč, ve kterém je potřeba mít nainstalovaný zásuvný modul. Tuto možnost využívá 99% uživatelů internetu v podobě Adobe Flash Player. Webová stránka pak obsahuje, podle specifikace, tag <object>, kterým importuje danou Flash aplikaci.

Příklad vložení aplikace do webové stránky:

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflas
h.cab#version=7,0,0,0" width="850" height="520" id="index" align="middle">
  <param name="allowScriptAccess" value="sameDomain" />
  <param name="movie" value="index.swf" />
  <param name="quality" value="high" />
  <param name="bgcolor" value="#f5f5f5" />
  <embed src="index.swf"
    quality="high"
    bgcolor="#f5f5f5"
    width="850"
    height="520"
    name="index"
    align="middle"
    allowScriptAccess="sameDomain"
    type="application/x-shockwave-flash"
    pluginspage="http://www.macromedia.com/go/getflashplayer" />
</object>
```

Další možností je spuštění aplikace jako desktopové, přímo ve Flash Playeru. V tomto ohledu se mohou vyskytnout problémy s funkčností aplikací, které byly vytvořeny pro web, ale spuštěny jako desktopové, nicméně se jedná pouze o problém komunikace se serverovou částí z důvodu bezpečnosti. Distribuci přehrávače je možné řešit přímo při exportu aplikace tak, že jej necháme připojit. Výsledná velikost aplikace se zvětší přibližně o 500kB, ale je možné ji přehrát na kterémkoli počítači se stejnou platformou (např. ve Windows s příponou .exe).

Samotný spustitelný soubor exportovaný z Flashe má příponu .swf. Jedná se o soubor aplikace, ve kterém je uložena veškerá funkčnost společně s designem výsledné aplikace.

2.4.3 ActionScript

ActionScript je programovací jazyk Flashe, který slouží k interaktivnímu propojení uživatele a flashové prezentace. S prvními verzemi Flashe se objevila potřeba interaktivně řídit animaci, kvůli tomu vznikl programovací jazyk ActionScript. Ten svou strukturou navazuje na v době vzniku již rozšířený a používaný JavaScript, který je používán k podobným účelům jako ActionScript dodnes. Tyto jazyky si navzájem přímo konkurují v podobě webových aplikací technologie AJAX a také právě Flash.

Již při vzniku ActionScriptu se počítalo s tím, že vývojáři budou ve značné míře znát i JavaScript, ovšem postupem vývoje se tyto dva jazyky stále více vzdalují.

První verze ActionScriptu měly za úkol jednoduché řízení animací (zastavení, přechod na určitý snímek apod.) a prvků uživatelského rozhraní výsledné aplikace jako jsou tlačítka.

Od verze 2.0 je již zavedena podpora objektově orientovaného programování, což jak již bylo dříve uvedeno, přineslo nárůst výkonu a širší programovací možnosti. Zároveň se značně rozšířily knihovny standardních funkcí a jejich možností, což je dokladem, že je ActionScript v dnešní době svébytným programovacím nástrojem.

Součástí této práce však není kompletní rozbor všech funkcí které Flash, potažmo ActionScript, přináší, proto zde budou uvedeny pro představu pouze vybrané zajímavé funkce, které nastíní široké možnosti práce s touto technologií.

Knihovna pro práci s médii je k tématu prezentace velice užitečná a obsahuje například:

- funkce pracující se vstupem webové kamery v reálném čase (např. nastavení snímkovací frekvence; práh pohybu obrazu, než se kamera automaticky aktivuje; kvalita přenášeného obrazu).
- funkce pro práci s mikrofonom (např. citlivost, práh šumu, práh hlasitosti okolních zvuků po aktivaci mikrofону v aplikaci)
- funkce pro práci se zvukem
- funkce pro práci s videem

Knihovna pro práci s používanými objekty scény zařizuje přístup k vlastnostem jednotlivých objektů jako je myš, klávesnice, textová pole, kontextová menu nebo tiskárna.

Další knihovny obsahují funkce pro práci s obrazem, jednoduššími grafickými filtry, geometrií, matematickými strukturami, externími zdroji, pro práci s textem, formátem XML, ID3 tagy MP3 souborů nebo formátování pomocí CSS.

2.4.4 Flex

Vzhledem k Flashi je Flex nástrojem s přístupem z opačné strany. Především se snaží podporovat vývojáře přehledným prostředím (v podobě komerčního Flex Builderu nebo jiného kompatibilního IDE) a deklarativním jazykem MXML na bázi XML pro popis vlastností a rozložení prvků grafického uživatelského rozhraní. Aplikační logiku reprezentuje nyní již třetí verze dobře propracovaného ActionScriptu. Flex dovoluje zapisovat vizuální efekty přímo pomocí MXML jazyka, což vývojářům přináší značné urychlení i ulehčení práce.

Nejde však pouze o vizuální efekty a líbivý vzhled aplikací jak je tomu u Flashe, podle obecného mínění především, jde o další ze způsobů tvorby RIA aplikací. Pomocí Flex jsou možnosti technologie rozšířeny tak aby bylo možné co nejefektivněji vytvářet kompletní projekty včetně rychlé komunikace se servery, přičemž sama firma Adobe tyto aktivity značně podporuje a za tímto účelem zřizuje množství specializovaných serverů.

Flex podporuje editaci, a tím i vývoj aplikací založených pouze na Action Scriptu, které jsou obvykle vytvářeny přímo ve Flashi, rozšiřuje tak uživatelský komfort při implementaci aplikační logiky. Vytváření nových aplikací se tak z programátorského úhlu pohledu srovnává s tvorbou v jiných programovacích jazycích.

2.4.5 Papervision engine

S příchodem ActionScript 2.0 a Flash Playeru 8 se zvýšil výkon Flash na tolik, že se i v této technologii začalo uvažovat o třetím rozměru. Vývojáři využili tehdejších možností a začalo vznikat mnoho komponent i celých webových prezentací, které využívaly zobrazení trojrozměrného prostoru. Šlo ale pouze o pseudo 3D prostor, který nebyl realizován standardními prostředky známými z 3D her a aplikací pro tvorbu, návrh či animace ve 3D. Realizované aplikace imitovaly prostor pomocí vrstev a jejich překreslování podle pořadí od pozorovatele. Situace by se dala přirovnat k posouvání papírů po stole, kterým patřičně upravujeme tvar a měníme jejich pořadí nad sebou tak, aby vizuální dojem lidskému oku připomínal prostorový vjem. Tato metoda byla velice výpočetně náročná, navíc nedovolovala tvorbu složitějších scén, protože bylo potřeba každou scénu vytvořit přímo v ActionScriptu.

Uvolnění ActionScriptu 3.0 společně s Flash Playerem 9 přinesl s sebou významnou optimalizaci spojenou s velkým nárůstem výkonu. V té době se začala masověji vyvíjet reálná simulace prostoru a Engine Papervision, který tvoří v roce 2009 nejpoužívanější API pro tvorbu 3D ve Flashi.

V současnosti je poslední vydanou verze 2.1.932 z prosince 2009, která stejně jako již mnoho předchozích, od verze 2.0, funguje pouze za použití Action Script 3.0.

Engine je postaven na otevřené platformě a jeho vývoj jde stále rychle kupředu. Současně se vyvíjí podpora importu více 3D formátů, společně s podporou importu kompletních animací, které některé formáty podporují. Rozšíření Enginu se zabývá i podporou simulace fyziky těles, materiály kapalin či podporou plynulé změny povrchu objektu pro animace postav, obličejů apod.

2.4.6 Collada formát

Collada (COLLABorative Design Activity) [9] formát s příponou .dae, je formát pro podporu přenosu 3D scén mezi různými platformami. Vznikl v roce 2004 jako snaha firmy Sony (brzy se připojily některé další firmy, např. Autodesk) vytvořit přenositelný, multiplatformní formát pro přenos komplexních 3D dat.

Jde o otevřený formát neziskového konsorcia Khronos Group, který využívá otevřeného XML standardu a má snahu sjednotit výstupy různých software řešení různých výrobců, dokonce sjednotit různé přístupy designérů.

Nově vzniklý formát měl v prvních verzích za úkol pokrýt potřeby přenosu pouze statických scén, ovšem brzy byly vyvíjeny nové verze, které již obsahují dynamické prvky nebo animace. Tento formát je navíc stále vyvíjen ve snaze obsáhnout i nejnovější technologie používané 3D editačními a vizualizačními nástroji, neboť ty se v dnešní době také velice dynamicky vyvíjí.

Od verze 1.4 byla do tohoto formátu přidána podpora fyziky. Cílem je, aby měli tvůrci obsahu možnost definovat různé fyzikální atributy použité ve scéně (např. tření). Zároveň je možné definovat fyzikální vlastnosti samotných objektů ve scéně. Toho lze dosáhnout díky přiřazení vybraných objektů do množiny těles, která jsou použita pro vizuální reprezentaci. Další funkce přináší podporu tzv. Ragdoll, kolize objemů objektů, fyzikální omezení mezi fyzickými objekty a nastavení globálních vlastností jako je gravitace nebo vítr.

Při vývoji byly také uvolněny překladače do tohoto formátu pro aplikace Maya (používá ColladaMaya), 3Ds Max (ColladaMax), Poser (v.7.0), LightWave 3D (version 9.5), Cinema 4D (MAXON), Softimage XSI, Side Effect's Houdini, MeshLab, CityScape, CityEngine, SketchUp, Blender, Modo a Strata 3D.

Snaha o rozšíření formátu je značná a uživateli žádaná. V současné době se však dá říct, že kompatibilita není na dobré úrovni. Otázkou kompatibility a přenositelnosti formátu Collada se zabývá samostatná kapitola 3.1.1.

3 Analýza prostředků

V této kapitole jsou diskutována možná řešení a prostředky použitelné pro tvorbu prezentací s pokročilými prvky. Pro výslednou aplikaci je potřeba vybrat vhodné z dostupných technologií. Při jejich výběru však, jak bylo zjištěno, není možné zcela spoléhat na informace uváděné výrobci softwaru, především v otázce kompatibility prostředků.

Celkově se na problému tvorby prezentací podílí větší množství technologií, které se všechny stále rychle vyvíjí. V konkurenčním prostředí výrobců tak dochází na jedné straně ke slučování technologií (např. přenosnost dat), tak aby jejich použití bylo co nejvíce efektivní, na straně druhé se snaží výrobci vytvářet i vlastní unikátní řešení (např. Flash proti Silverlight), zřejmě ve snaze se od konkurence odlišit a v případě zájmu tak získat zákazníky, kteří nemohou využívat prostředků konkurence. Otázka kompatibility je tak stále aktuální problém.

3.1 Problém kompatibility

Při zkoumání různých možností tvorby prezentací s pokročilými prvky vzniklo několik možných řešení, která se mohou propojit při potřebě vzniku komplexnějšího díla, které by mělo být použitelné na větším množství zařízení a platform.

V takovém případě je problém vzájemné použitelnosti některých technologií výraznější a obecně se může zvyšovat s nárůstem počtu použitých prostředků. Vzhledem ke stálému vývoji je navíc tato otázka otevřená a v případě vzniku nových řešení do budoucna není vyloučeno, naopak je spíše pravděpodobné, že se tak vždy bude v důsledku jednat o velice důležitý faktor pro výběr určitého řešení.

Příkladem může být vznik nové verze formátu, která, aby splnila narůstající potřeby uživatelů, přinese taková řešení, jež nebudou zpětně kompatibilní. V takovém případě však vzniká problém i u technologií, které s daným formátem pracují. Celkově tak vzniká potřeba vše upravit do funkčního stavu a výsledkem je tak více nových verzí, což může být problém při vývoji nových řešení, která nejsou dostatečně popsána.

Z technologií popsaných v kapitole 2 je potřeba vybrat pro realizaci takové, které budou nejlépe vyhovovat zvolenému cíli. V následujících kapitolách je uveden výběr nejdůležitějších kombinací použitelných prostředků vzhledem k vzájemné kompatibilitě, bez ohledu na konečný výběr pro realizaci. Zároveň jsou však uvedeny pouze prostředky, které přichází v úvahu v případě reálné implementace takové, že je možné zvolený cíl realizovat bez nutnosti vývoje vlastních komplikovaných prostředků.

3.1.1 Převod do formátu Collada (.dae)

Výstupem 3D scanneru může být několik různých formátů. Při volbě zobrazení konečného modelu v prezentaci, založené na technologii Flash, je však potřebný formát Collada (.dae), který implicitně součástí softwaru používaného při skenování není. Převod jednotlivých formátů je možný pomocí konvertoru, tedy aplikace, která překládá jednotlivé formáty.

V dnešní době existuje několik konvertorů do formátu Collada, z nichž většina je součástí větších aplikací, 3D editorů. Zde však vzniká patrně největší problém. V současnosti je Collada 1.5 nejnovější verzí, která pouze rozšiřuje verzi 1.4.1. Verze, která je však používána v praxi nejvíce, je Collada 1.4.0, ta ale není s novějšími kompatibilní.

V převodu prostorového modelu do formátu Collada byly testovány aplikace: 3Ds MAX 2009, Blender 2.5.2, Blender 2.492 a Maya 2010. Aplikace Blender 2.492 exportovala formát .dae pomocí zásuvného modulu firmy Illusoft, který navíc dovoluje export i do verze Collada 1.3.1.

		export Collada (.dae)			
		Blender 2.492	Blender 2.5.2	3Ds MAX 2009	Maya 2010
načtení	Blender 2.492	ANO	NE	NE	NE
	Blender 2.5.2	NE	ANO	NE	ANO
	3Ds MAX 2009	ANO	NE	ANO	ANO
	Maya 2010	ANO	NE	ANO	ANO

Tabulka 1: Schopnost aplikací načíst formát .dae

V předchozí tabulce je vyznačena schopnost přenosu formátu mezi zvolenými aplikacemi, kde je vidět, že ačkoliv formát vznikl za účelem přenosu dat, není tento cíl zcela naplněn. Přenášaná data navíc neobsahovala pokročilé vlastnosti fyziky či animací, šlo pouze o přenos 3D objektu.

Vzhledem k existenci různých verzí formátu Collada a skutečnosti, že aplikace neumožňují explicitní výběr, je vhodné zjistit, do jaké verze daný program naskenovaný model ukládá, což shrnuje následující tabulka:

aplikace	verze Collada (.dae)
Blender 2.492	1.4.0
Blender 2.5.2	1.4.1
3Ds MAX 2009	1.4.0
Maya 2010	1.4.0

Tabulka 2: Verze exportovaného formátu Collada

Z předchozích informací je možné odvodit, že problém kompatibility je v tomto případě značně neprůhledný. Práce s formátem je tak možná, ale není příliš vhodná pro realizaci větších projektů, ve kterých by bylo dynamicky zapojeno více subjektů. Situace s výhledem do budoucna je také nejistá.

3.1.2 Formát pro Papervision engine

Nejrozšířenějším nástrojem pro zobrazování prostorového obsahu v prostředí Flash je Papervision engine, který je stručně popsán v kapitole 2.4.5. Toto rozšíření obsahuje velké množství tříd pro celkovou práci s 3D obsahem. V této kapitole se jedná pouze o snahu osvětlit problém verze Papervision 2.0 a vyšší, protože starší verze, ačkoliv je na ni možné odkazy stále nalézt, se v dnešní době nepoužívá.

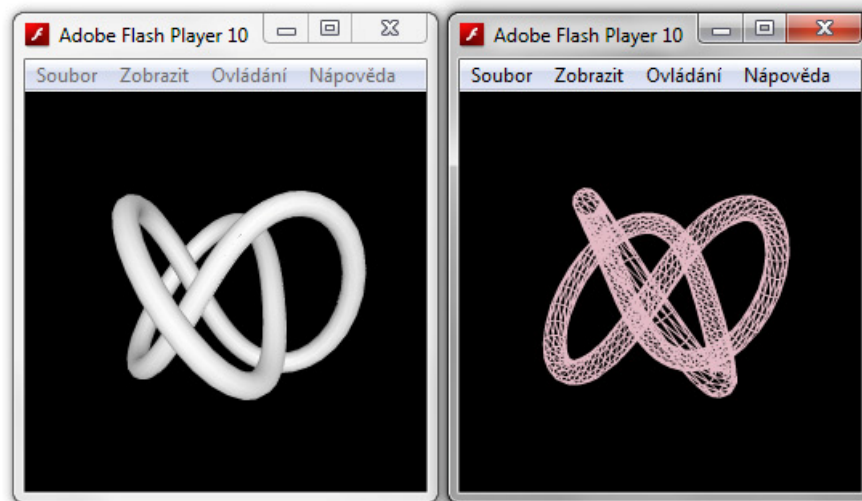
Pro potřeby importu modelů je nejdůležitější balík `org.papervision3d.objects.parsers`, který obsahuje třídy pro čtení formátů `.ase`, `.kmz`, `.3ds`, `.MD2` a `.dae`.

Dokladem toho, že situace s formátem Collada není úplně jednoduchá je fakt, že pro překlad souborů s příponou `.dae` jsou v balíku 4 třídy:

- Collada - pro zpracování Collada scén verze 1.4.0. Tato třída navíc obsahuje i doporučení pro nastavení exportu scén z aplikací 3Ds MAX a Maya, které se liší.
- DAE - pro zpracování Collada formátu verze 1.4.1, tedy rozšíření o možnost importu animací.
- SketchupCollada - třída téměř shodná s Collada. Liší se pouze v nastavení základního materiálu a použití oboustranných materiálů (SketchupCollada povoluje i jednostranné).
- Sketchup - kombinace SketchupCollada a třídy KMZ.

Teoreticky by na základě předchozího nemělo být problematické prostorový obsah zobrazit. V praxi je však situace jiná a vyskytuje se značný problém, především v zobrazení materiálů.

Každý parser vyžaduje pro uložení dat objekt speciálního typu, který je v dané třídě definován. Pro import Collada 1.4.0 vypadá jednoduchá část kódu pro zobrazení objektu tak jak je ukázáno v příloze A.1. V kódu jsou zvýrazněny části, které se týkají způsobu práce s objekty pro import a zobrazení. Kód se pro verzi Collada 1.4.1, jak je doloženo v příloze A.2, liší pouze v použitých typech objektů, to má ovšem v důsledku vliv při renderování scény. V případě A.2 je objekt zobrazen korektně s plným materiálem základní světle šedé barvy. V případě A.1 je ale zobrazen pouze drátěný model objektu. Situaci dokresluje následující obrázek 3.1.



Obrázek 3.1: Zobrazení formátu Collada 1.4.1 (vlevo) a Collada 1.4.0 (vpravo)

Řešení, které produkuje poměrně malou poruchovost, je v použití vhodného materiálu. V tomto případě jde o využití materialu typu `bitmapFileMaterial`. Jde o nanesení zvoleného externího obrázku na plášť zobrazovaného modelu. Model však musí obsahovat koordináty pro mapování textury (obrázku), jinak je textura nanesena automaticky, což může být u jednodušších těles žádaná vlastnost, u složitějších těles ale nefunguje.

3.1.3 Video a Flash ve VRML

Virtuální scény vytvořené v jazyce VRML mohou být velice komplexní. Nedovolují sice interakci více uživatelů, ale to je, s ohledem na potřeby prezentace, vlastnost okrajová. Na druhou stranu je v jazyce VRML možné svět vytvářet pomocí značné škály nástrojů. Jedním z pokročilějších je využití videa. K videu se zde přistupuje podobně jako k jinému multimediálnímu obsahu. Stejně jako je možné texturovat prostorové objekty zvolenými obrázky, je možné texturovat i pomocí videa. Ve skutečnosti je to velice jednoduché a k nanesení animované textury, například na krychli, stačí následující krátký kód:

```
Shape {
  appearance      Appearance {
    material       Material {
    }
    #textura s videem přehrávaná ve smyčce
    texture MovieTexture {
      url          "video.mpg"
      loop         TRUE
      startTime     0
      stopTime      -1
    }
  }

  geometry Box {
  }
}
```

Video může mít přiřazeno více vlastností, než je v ukázce, omezení ale existuje v otázce použitého formátu videa. Podporována jsou standartně pouze videa komprimovaná metodou MPEG 1, MPEG 2 nebo MPEG 4. Pro podporu jiných formátů by uživatel musel použít prohlížeč obsahující speciální rozšíření.

Prohlížečů VRML obsahu celkově neexistuje velké množství. Nejpoužívanějším je prohlížeč Cortona 3D viewer, vedle něj je doporučován freeWRL nebo Vivaty Player, který je primárně určen pro zobrazování X3D, ale VRML zobrazí také. Test prohlížečů a jejich schopnosti zobrazení video textury ukázal, že freeWRL není schopen video vůbec zobrazit. Ostatní programy přehrávání videa podporují, i když nároky na hardware se různí, stejně jako kvalita přehrávání. Obecně se nedoporučuje používat jako texturu větší videa (myšleno objemem dat).

Příkladem použitelné textury může být video o rozměru 320 x 240 pixelů, komprimované metodou MPEG 4, které má celkovou velikost 1,4 MB. Nevhodným příkladem použití je textura videa o rozměru 720 x 576 pixelů o celkovém objemu dat 82MB. V tomto ohledu je vhodné připomenout původní účel VRML, tedy možnost zobrazení virtuálního světa na internetu. Vzhledem k tomu, že je při zobrazování potřeba vždy načíst všechny objekty a textury, je nutné dbát na to, aby celkový objem dat scén příliš nenarůstal.

Cortona 3D při přehrávání využívá internetového prohlížeče, což má v důsledku vliv na výkon a scéna obsahující video byla obnovována frekvencí mezi 16 až 19 fps (snímky za sekundu) což původní video s 25 fps zpomaluje a způsobuje zadržávání. Naproti tomu Vivaty Player přehrává video správnou rychlostí, obraz je navíc zřetelně ostřejší jak ukazují následující obrázky 3.2 a 3.3.

Vlastnost, která by mohla být v oblasti prezentací velice vítaná je propojení interaktivního, dalo by se říct plochého (co se prostoru týče) prostředí Flash a virtuálního prostoru vytvořeného pomocí VRML. Při zkoumání této možnosti bylo zjištěno, že pravděpodobně jediným prohlížečem VRML, který podporuje použití Flash jako animované textury je Cortona 3D viewer. Bohužel, toto řešení funguje pouze online, protože rozšíření, které dovoluje Flash zobrazit se musí načítat z internetových stránek firmy Parallelgraphics.



Obrázek 3.2: Ostrost videa v přehrávači VRML Cortona 3D



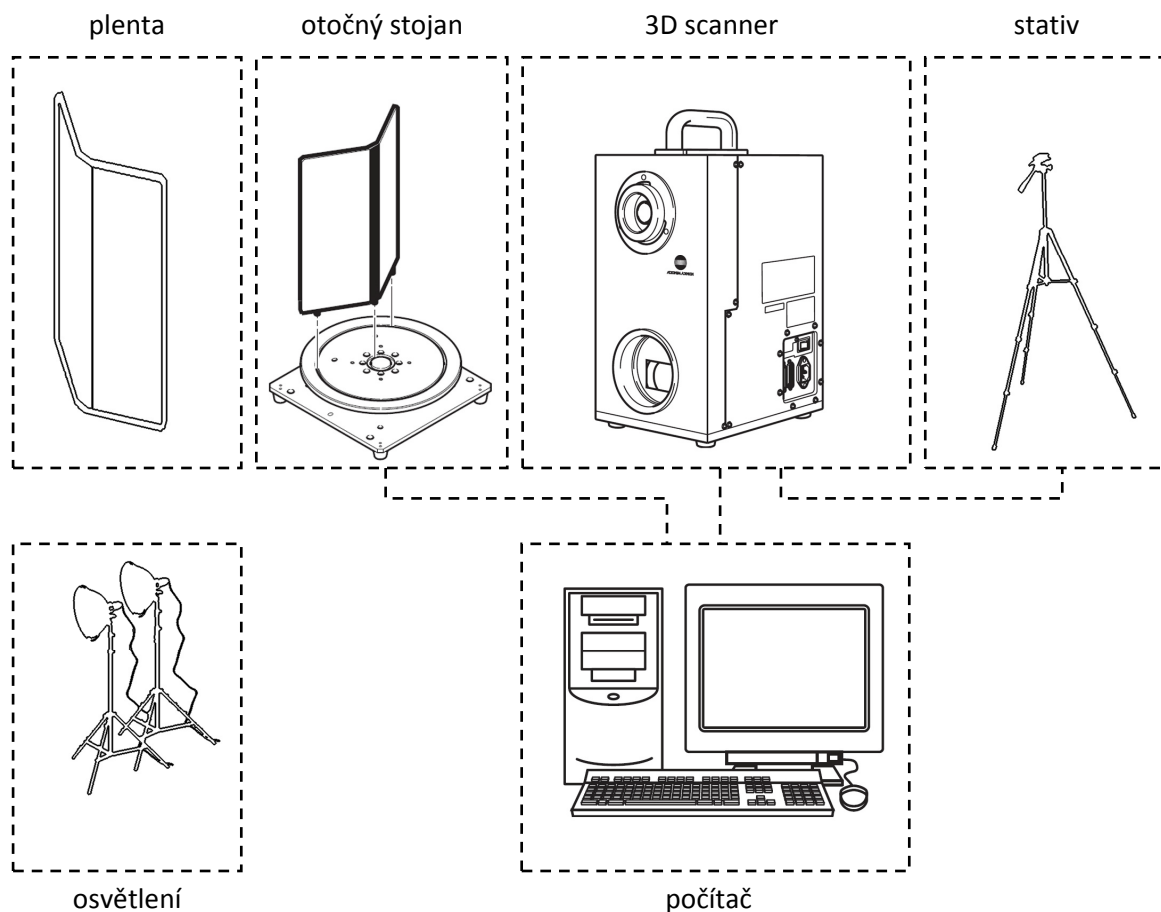
Obrázek 3.3: Ostrost videa v přehrávači VRML Vivaty Player

3.2 Použití 3D scanneru

Možností, jak vytvořit prostorový model, který by byl použitelný pro prezentaci potřebných vlastností, existuje několik. 3D scanner je zařízení využívané v současnosti v největší míře spíše k jiným účelům než je pořizování modelů do prezentací (hlavně reverzní inženýrství a ochrana kulturního dědictví). Tento stav vychází z několika skutečností.

Pravděpodobně nejzásadnějším faktorem, který hraje proti masovějšímu využití 3D scanneru, jsou pořizovací náklady. Samotný přístroj je finančně náročné a ačkoliv je scanner uzpůsoben tak, že je možné jej použít samostatně bez jakéhokoli jiného vybavení, jeho reálné možnosti jsou tak značně sníženy. Ve skutečnosti je k využití celého potenciálu přístroje potřeba řada

zařízení, které tvoří pracovní stanici a prostor pro skenování. Následující obrázek ukazuje standartně používané vybavení.



Obrázek 3.4: Systém zařízení používaný pro 3D skenování

Dalšími faktory omezující použití jsou materiály a tvary skenovaných objektů. Technické parametry 3D scanneru Konica Minolta Vivid 910 neobsahují přesnější specifikaci, jaké objekty je možné skenovat kromě, hrubého určení jejich velikostí. Z tohoto důvodu byla provedena řada testů, které by měly poskytovat vodítko pro efektivní práci se zařízením.

Doporučení, hodnocení a případné závěry vznikly na základě reálné zkušenosti s používáním přístroje v laboratoři. Laboratoř nebyla při testování scanneru ještě plně vybavena a samotné zařízení bylo instalováno pouze krátce. To však v důsledku mohlo mít i kladný vliv v procesu testování, neboť tak bylo možné zjišťovat vlastnosti při reálném či obecném použití.

3.2.1 Prostředí

Prostředím je myšlen prostor, ve kterém je prováděno skenování objektů. Při studiu současné situace bylo zjištěno, že je 3D scanner velmi často používán v exteriérech např. u archeologických vykopávek, ale i v interiérech, ve skladu archeologických předmětů apod. Ve všech případech je ale pro kvalitní výsledky třeba vytvořit vhodné prostředí.

Vhodné prostředí by mělo splňovat několik základních vlastností tak, aby mohl být využit celkový potenciál skenovacího zařízení:

- možnost automatického zaostření a odhad intenzity laseru
- produkce nízkého počtu nesprávných bodů
- produkce kvalitní textury.

Automatické zaostření a odhad intenzity laseru jsou vlastnosti, které urychlují a zjednodušují celý proces snímání objektu. Bez této možnosti by bylo potřeba nastavovat pro každý objekt vlastní hodnoty ostřicí vzdálenosti a intenzity laseru. V případě, že je pořizován pouze jeden snímek, není časová úspora příliš výrazná, ale v případě, že pořizujeme všesměrový model za použití otočného podstavce, je časová úspora značná. Problém je demonstrován na následujících příkladech.

Objekt je umístěn na otočném podstavci, který umožňuje 360° snímání. Zařízení je nastaveno tak, že jsou snímky pořizovány po 60°, celkově jde tedy o pořízení 5 snímků. Kvalita snímání je nastavena na nejvyšší možnou, včetně snímání barevné textury povrchu. Průměrné časy snímání objektu shrnuje následující tabulka:

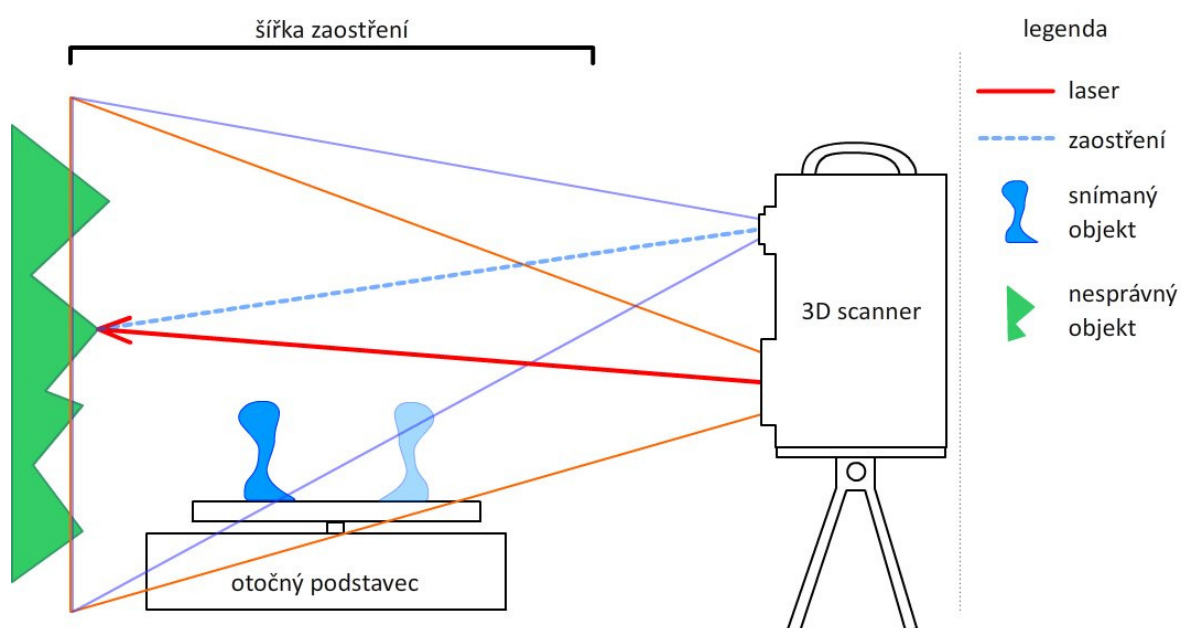
proces	průměrný čas (sec)
zaostření a odhad intenzity laseru	3
snímání objektu (režim fine)	3 * 3
jedno pootočení podstavce (60°)	5
vrácení podstavce do původní polohy (300°)	5 * 5
celková doba snímání	62

Tabulka 3: Průměrné časy snímání v sekundách

Chybná funkce zařízení může být způsobena následujícími situacemi nebo jejich kombinacemi:

- do snímaného prostoru zasahuje nesprávný objekt, který má výraznější strukturu (např. obrázek na zdi), než objekt snímáný. Automatika v tomto případě zaostří na nesprávný objekt, snímání proběhne celé, avšak výsledek je chybný.
- snímáný objekt je umístěn výrazně mimo osu otáčení podstavce. To způsobí, že první snímky mohou být pořízeny v pořádku. Při automatickém přeastřování však může dojít k chybě a snímání je pozastaveno až do provedení manuální korekce.
- u malých objektů (cca. méně než 2x2 cm) často dochází k chybnému měření pro určení intenzity laseru i při správném zaostření. Laser je pak nastaven příliš silný a snímání vytvoří model chybný nebo dokonce žádný.

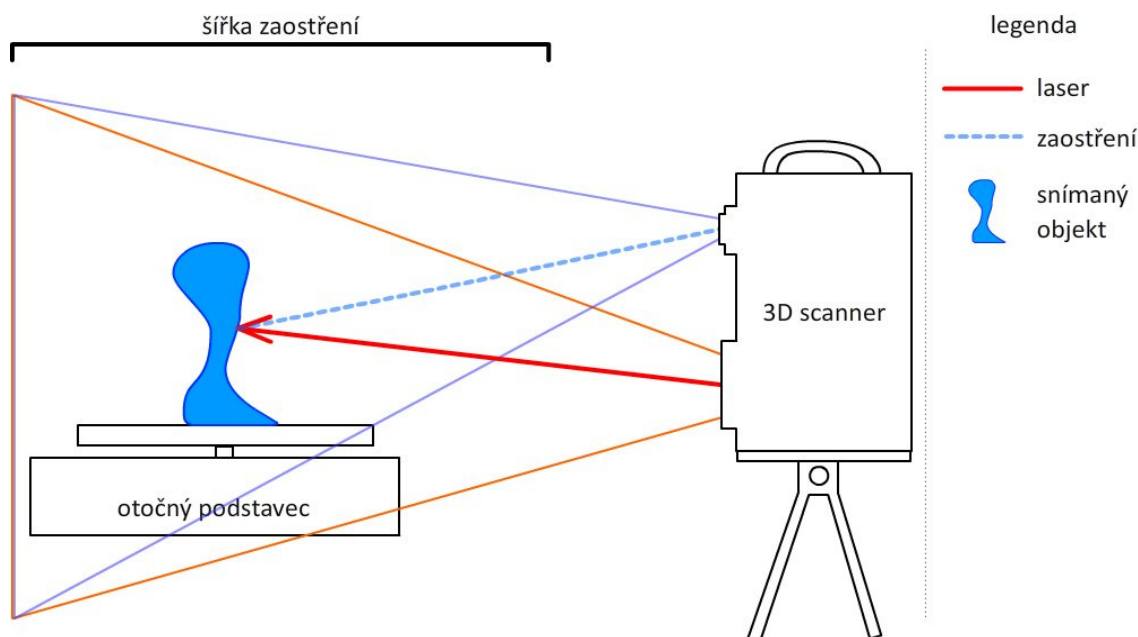
Následující obrázek představuje vzniklou situaci:



Obrázek 3.5: Diagram prostředí s vysokou chybovostí v automatickém režimu snímání

V případě, že automatický režim selže, je možné použít režim manuální nebo scénu upravit tak, aby bylo odstraněno co nejvíce faktorů, které chyby způsobují:

- předměty zasahující do snímaného prostoru, který tvoří do vzdálenosti od scanneru cca 2,5 metru, je možné odsunout nebo zakrýt jednolitým materiálem. V praxi stačilo umístit před nesprávný objekt plachtu nebo plátno.
- změnou umístění předmětu a podstavce v prostoru vůči scanneru tak, aby snímaný objekt nevybočoval z osy rotace podstavce
- výměnou objektivu je možné docílit toho, aby předmět zájmu zaujímal v pracovním prostoru přístroje co největší místo



Obrázek 3.6: Situace po odstranění faktorů způsobujících chyby automatického režimu snímání

Počet nesprávných bodů komplikuje tvorbu výsledného virtuálního modelu tím více, čím více jich obsahuje. Nesprávnými či nepotřebnými body jsou myšlena místa, která jsou nasnímána a zahrnuta do podoby vznikajícího digitálního 3D modelu, ale ve skutečnosti nekorrespondují s fyzickým objektem, o který máme zájem. Vznikají v místech, kde laser nasnímá povrch jiného než požadovaného tělesa nebo vytvoří povrch v místě, kde fyzicky žádný není. To ve výsledku komplikuje případné automatické složení celého modelu. Software pro práci s naskenovanými daty obsahuje nástroje pro odstranění nepotřebných míst, ale jde o manuální činnost, která může vyžadovat značné množství času.

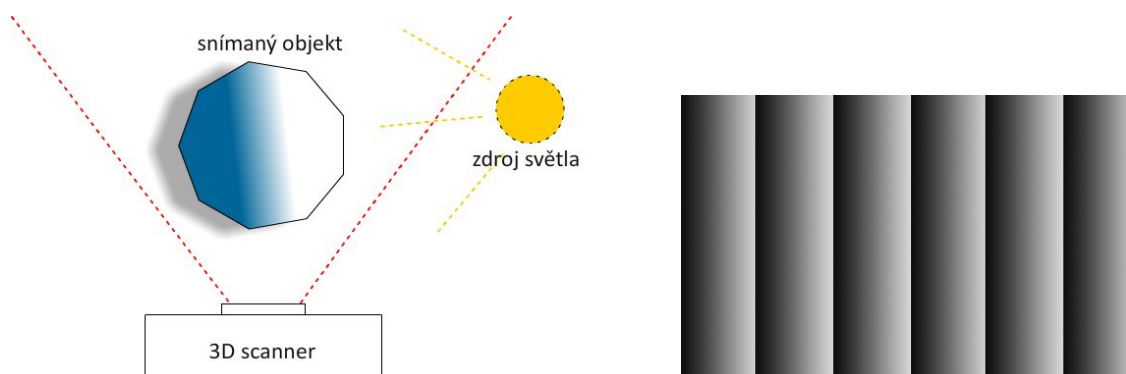
Problém vytváření nepotřebných bodů úzce souvisí s předchozím problémem automatického zaostření. V případě, že je přístroj správně zaostřen a také vhodně zvolí intenzitu laseru, měl by být počet chybných bodů minimální. K tomu přispívá například i povrchová úprava otočného podstavce.

Kvalitní textura povrchu je pro řadu objektů klíčová a u mnohých má textura dokonce zásadnější význam než samotný tvar předmětu, i když se tyto vlastnosti do značné míry doplňují.

Ke snímání textury se však váží specifické problémy. Jde o informaci o barvě, která je zásadním způsobem ovlivňována světlem. Přesněji řečeno, stačí se zaměřit na osvětlení prostoru a snímaného objektu. V tomto ohledu je potřeba se zajímat o směr, intenzitu a barvu osvětlení.

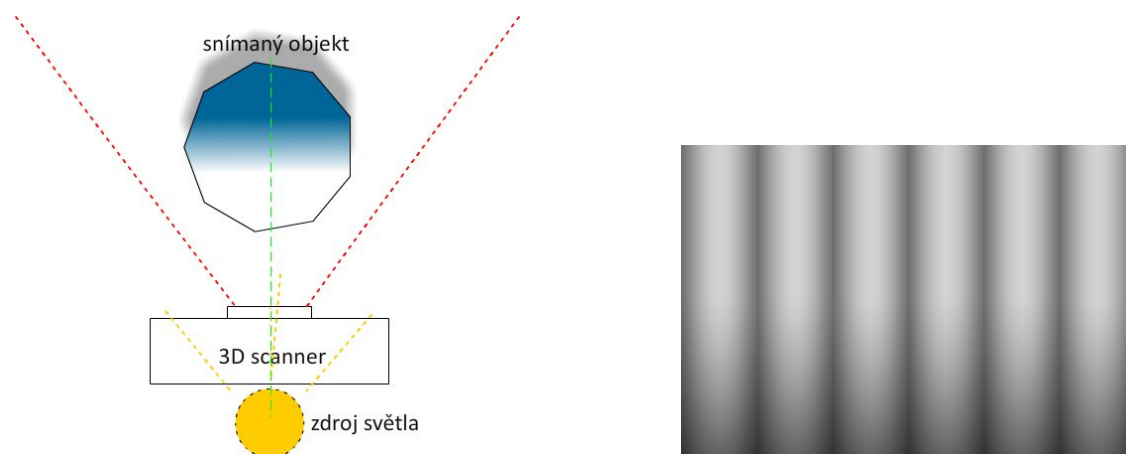
Základní nasvětlení scény využívá přírodního světla nebo např. světel v místnosti, která nejsou primárně určena pro osvětlení scény. V takto nasvětleném prostoru bývá intenzita osvětlení většinou do 500 lx (lux), což je hodnota, která je pro snímání vhodná. Příkladem nasvětlení, které není vhodné a znemožňuje skenování, je přímé osvětlení sluncem. Ani při nejvyšší zvolené intenzitě laseru pak není možné potřebný objekt nasnímat. Většina pokusů skončí chybovou zprávou, která upozorňuje na

fakt, že nebyla přístrojem zjištěna žádná data vhodná k uložení, nebo jsou výsledkem pouze malé fragmenty původního objektu (např. v místech stínu). Avšak i při vhodné intenzitě osvětlení může docházet při základním nasvětlení k vytvoření textury, která není vhodná či použitelná.



Obrázek 3.7: Schéma základního nasvětlení objektu (vlevo) a textura (vpravo)

Obrázek 3.7 ukazuje problém, který vzniká při použití jediného zdroje světla umístěného na straně od objektu vůči ose přístroje. Textura pořízená snímáním po 60° je ve výsledku složením jednotlivých obrazů, obsahujících přechod barvy od tmavé (zastíněné) po světlou (nasvícenou). Možnost, jak nastavit scénu s jedním zdrojem světla tak, aby byl výsledek co nejlepší, je umístění zdroje světla na pracovní ose přístroje, jak ukazuje obrázek 3.8. Změnění úhlu mezi jednotlivými snímky sice prodlužuje dobu skenování, ale má pozitivní vliv na výsledek, který je spočten na základě většího objemu dat. Zároveň se problém osvětlení jednotlivých snímků rozmělní tak, že nemusí být na výsledné textuře patrný.



Obrázek 3.8: Zdroj světla v ose scanneru (vlevo) a textura (vpravo)

Přidáním druhého zdroje světla a jeho vhodným umístěním se situace také výrazně zlepší. V praxi se často používají tři zdroje světla. Pracovní prostor je navíc od okolí oddělen jednoduším materiálem (tkanina, papír) s difúzní vlastností. Rozptýlené světlo z několika zdrojů vytváří prostředí s neznatelnými či velmi měkkými stíny, které je vhodné pro snímání potřebné textury. Příklad vytvoření ideálního prostředí v praxi je možné vidět v příloze A.4.

V části 3.2.1 byla uvedena praktická vodítka pro práci s bezkontaktním laserovým 3D scannerem, přičemž se teoreticky předpokládá, že vzniklá doporučení mohou být obecně použita pro všechny přístroje stejného typu, které se tak budou lišit pouze v konkrétních hodnotách.

3.2.2 Nastavení přístroje

Součástí přístroje Konica Minolta Vivid 910 je podrobný manuál popisující postup, jak se zařízením pracovat a jaké jsou možnosti jeho nastavení. Tato část se zaměří na praktickou zkoušku hlavních parametrů a teoretická východiska vyplývající z testů. Uvedené hodnoty byly zjišťovány na konkrétním přístroji. Pro různé přístroje stejného typu nejsou dané hodnoty obecně shodné, existuje však předpoklad, že zjištěné trendy hodnot je možné zobecnit na základě fyzikální podstaty fungování zařízení, která je u daného typu přístrojů stejná.

Distance - vzdálenost

Popis: Distance určuje vzdálenost objektu od přístroje.

Hodnoty: 600-2500 mm

Použití: Při zapnutí funkce automatického ostření (AF) se hodnota Distance vypisuje, takže je možné ji průběžně kontrolovat, ale nastavuje se automaticky bez možnosti ji měnit. Při výběru manuálního nastavení (v zatrhávacím políčku) se AF vypne a přístroj je zaostřen do vzdálenosti odpovídající zvolené hodnotě.

Hodnota vzdálenosti je pouze přibližná, na přesnost měření laserem, tedy i snímání prostorového modelu, nemá vliv. Určuje ale střední rovinu zaostření optického členu pro získání textury. Hloubka ostrosti následně také napomáhá k přibližnému určení pracovního prostoru laseru. Díky tomu je přístroj schopen odstranit z prostorových dat ta, která jsou pravděpodobně chybná.

Intensity - intenzita laseru

Popis: Intensity určuje intenzitu laseru při snímání

Hodnoty: LD: 0-255; Gain: 0-7

Použití: Funkce automatického nastavení intenzity laseru je v základním nastavení zapnuta. Hodnoty LD určující intenzitu laseru v nedefinovaných jednotkách je možné volit v 256 hodnotách. Nastavení odpovídá poměru intenzity laseru, kde hodnota 0 představuje nejmenší intenzitu. Nejvyšší intenzita je nastavena při hodnotách LD = 255 a současně Gain 7. Při jakémkoli nastavení hodnot Gain je LD automaticky 255, jde o přírůstek k maximální hodnotě LD.

Jak bylo možné pozorovat při snímání, automatické hodnoty se nejčastěji pohybovaly v rozmezí 10 - 25. Konkrétní hodnoty jsou však odvislé vždy od vlastností snímané scény (intenzita osvětlení, vzdálenost nebo barva předmětu snímání). Proto byly v reálném testovacím prostředí ověřovány pouze teoretické trendy pro nastavení hodnot.

Bylo pozorováno, že při vzdalování předmětu od snímače se hodnota intenzity zvyšuje. Při snižování intenzity osvětlení předmětu roste intenzita laseru, což koresponduje se světlostí barvy předmětu, kdy je potřeba nastavit intenzitu laseru na vyšší hodnotu pro tmavší barvu, naopak světlé materiály vyžadují hodnoty nižší.

Příkladem může být bílý pórovitý kámen (pemza), u kterého musela být hodnota intenzity laseru nastavena ručně na hodnotu 7-12 (kdy byla kvalita dat nejlepší), tedy nižší než automatickou (17). Při volbě intenzity menší než 5 však už nebyla nasnímana žádná data, přičemž scéna byla nasvětlena pouze plným kancelářským stropním osvětlením.

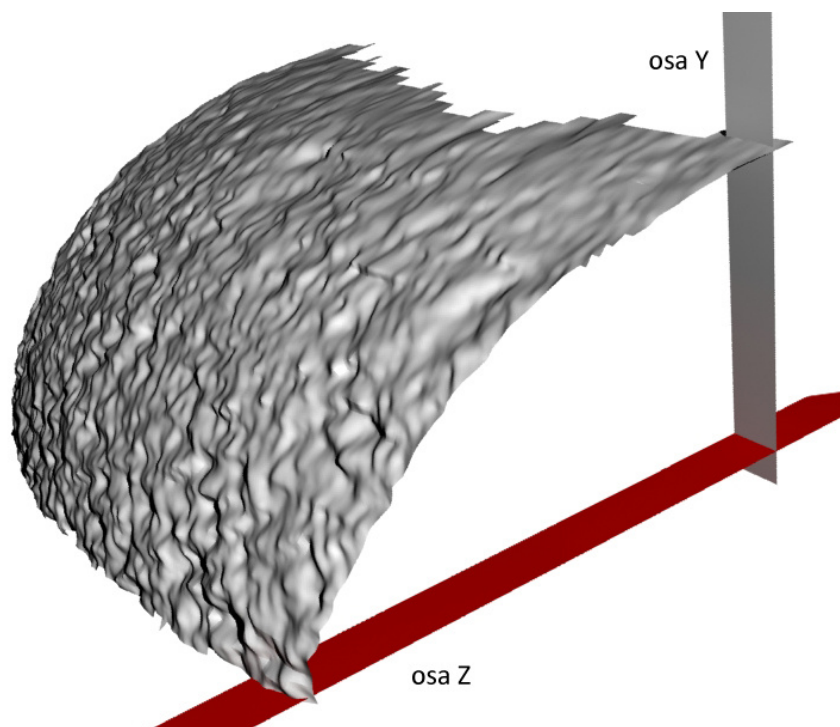
Rotation Step - Krok rotace

Popis: Krok rotace určuje úhel o který se pootočí rotační podstavec

Hodnoty: 0° - 360°

Použití: Nastavení je možné volit v předvolených krocích 90° , 60° , vlastní nastavení (vlastní hodnota pevného kroku) a ve volném režimu, kdy je hodnota otočení určena manuálně pro každý snímek (horizontálním posuvníkem).

Volba kratšího kroku snímání prodlužuje čas potřebný ke skenování předmětu, kvalita výstupu je však vyšší. Obecně je kvalita okrajových částí předmětů (především rotačních těles) v jednotlivých snímcích nižší, což dokládá následující obrázek 3.9. Nepřesnosti vznikají v blízkosti místa tečny paprsku laseru s povrchem předmětu. Z toho důvodu je skenování prováděno z různých směrů a v daném místě se při složení jednotlivých částí nachází větší množství dat pro výpočet výsledného povrchu.



Obrázek 3.9: Fragment povrchu předmětu s viditelnou změnou přesnosti na horním okraji

Při volbě velkého úhlu otočení, 90° a více, vzniká z důvodu nedostatku kvalitních dat problém v místě, kde by se jednotlivé povrchy měly spojovat. Povrch výsledného modelu a jeho přesnost se v tomto ohledu může výrazně lišit od skutečnosti, případně typických hodnot přesnosti měření.

Dynamic Range Expansion Mode

Popis: Režim dynamické intenzity laseru

Hodnoty: Zapnuto, Vypnuto

Použití: Funkci dynamické intenzity laseru je možné použít pouze v režimu Fine (kvalitnější snímání). Jde o trojí snímání namísto jednoho, pokaždé s jinou intenzitou laseru. V důsledku je výsledný virtuální model přesnější. Zlepšení kvality povrchu je znatelné především v místech, kde je fyzický povrch předmětu např. lesklejší, více nasvícený nebo jinak odlišný od zbývajících částí povrchu.

V praxi tato funkce vylepšuje výsledky snímání především v místech vzniku děr ve virtuálním povrchu. Těch je pak ve většině případů méně a jsou menší, což také výrazně zlepšuje účinnost jejich automatického odstraňování.

3.3 3D v prezentaci

Někteří odborníci se shodují [10], že v následujících pěti letech by měly internetové prezentace dostávat prostorovou podobu. V tomto ohledu informace korespondují i s tendencemi firem Mozilla a Google společně s výrazným rozvojem techniky 3D zobrazovačů a zájmem veřejnosti.

Při tvorbě pokročilých prezentací s prvky 3D existují dvě hlavní části, v nichž je jejich využití možné. Každá prezentace je složena z uživatelského rozhraní pro ovládání prezentace a obsahu, který je prezentován. Z širšího pohledu je prezentace, stejně jako jakékoli jiné dílo, tvořena obsahem a formou, což jsou části neoddělitelné, které by ve výsledku měly tvořit ucelený svazek. Z tohoto pohledu je nutné vnímat uživatelské rozhraní jako část prezentace, která by měla mít také svůj vlastní obsah a formu, vhodně korespondující s vlastním obsahem celé prezentace.

3.3.1 3D uživatelská rozhraní

Uživatelské rozhraní v prostoru je možné si představit v hlavních dvou směrech, ve kterých se v současnosti v praxi používá.

Prvním využitím je použití prostorových prvků pro navigaci uživatele. Takto vytvořený vizuální styl může výsledek výrazně odlišit od jiných 2D řešení. Příkladem může být situace na současném internetu, kde zastoupení webových stránek s prostorovým uživatelským rozhraním tvoří nepoměrně malé procento. Je to dáno pravděpodobně tím, že použití dostupných technologií je poměrně složitější než tvorba standardního webového obsahu (neexistují automatizační nástroje jako

v případě tvorby blogů a jiných webových stránek), vykreslování prostoru je náročnější na hardware počítače a také to, že uživateli nejčastěji hledaný obsah byl doposud text a video [11]. U textu (delší texty, zprávy, sdělení) nelze předpokládat ani do budoucna, že by vznikla snaha jej zobrazovat prostorově. Otázka videa je rozebrána v kapitole 3.4.

Pro tvorbu prostorového uživatelského rozhraní, které je ale napojeno na 2D obsah, se nejvíce používá technologie Flash. 3D prostoru je v tomto směru využíváno především pro tvorbu menu pro výběr obrázků, videí nebo rozcestníků prezentací.



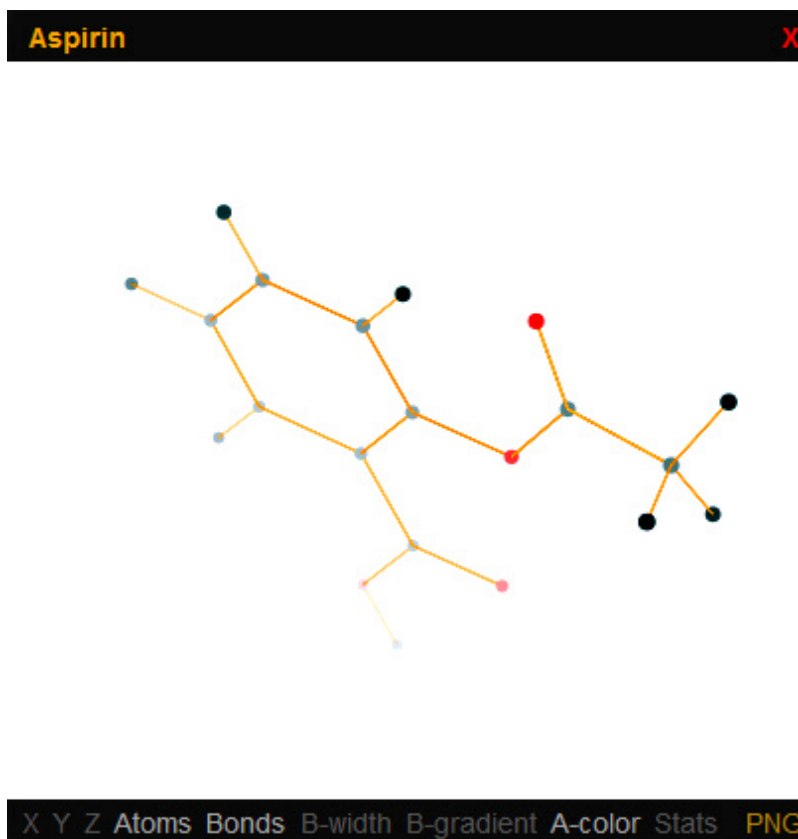
Obrázek 3.10: Ukázka 3D rozcestníku webových stránek <http://www.screenvader.com>

3.3.2 3D obsahová část prezentace

3D obsah prezentace může představovat nekonečnou škálu témat a objektů. Předmět prezentace není téměř ničím omezen, např. u zobrazování fyzicky reálných objektů sice zůstává faktem, že vše je pouze zjednodušeným modelem skutečnosti. Vhodná míra abstrakce či zjednodušení je ale také v určitých případech účinným prostředkem prospívajícím účelu prezentace.

Využití 3D je vhodné především tam, kde je snaha zobrazit či vysvětlit určité prostorové vlastnosti. Nemusí však jít pouze o fyzické vlastnosti jako tvar povrchu, umístění apod. Při vhodné volbě reprezentace je možné zobrazit i nemateriální vlastnosti jako jsou např. vztahy, souvislosti vazby apod.

Na obrázku 3.11 je vidět jeden z příkladů jak je možné využít prostorového zobrazení. Jde o aplikaci, ve které je možné prohlížet prostorové složení molekul vybraných látek. Dalším příkladem je zobrazení určitého předmětu, konkrétně zde archeologického nálezů. Obrázek 3.12 je vypůjčen z archeologického muzea The Virtual Hampson Museum (<http://hampsonmuseum.cast.uark.edu>), kde je možné si všechny předměty virtuálně prohlédnout. Dalším a zřejmě nejčastějším tématem 3D prezentací jsou virtuální prohlídky konkrétních míst, náměstí, hradů, budov apod.



Obrázek 3.11: Ukázka chemického složení Aspirinu v prostoru
(z aplikace na <http://alteredqualia.com/canvasmol>)



Obrázek 3.12: Ukázka jednoho z předmětů z muzea The Virtual Hampson Museum

Při hledání technologií, které se dnes používají pro prostorovou prezentaci bylo zjištěno, že pravděpodobně nejpoužívanější je v této oblasti Flash, je to dáno nejspíš velkým počtem lidí, kteří se zabývají vývojem a také tím, že odlišná řešení vyžadující, stejně jako Flash, zásuvný modul v internetovém prohlížeči nejsou tak rozšířená. Situace na českých webových stránkách je ještě o poznání méně přehledná, než jinde ve světě. Ačkoli je 3D obsahu využíváno v podobné míře, použité technologie se liší téměř případ od případu a množství firem používá vlastní zásuvné moduly, které je nutné instalovat.

3.4 Video

Video je technologie pro zaznamenávání, přehrávání a přenos pohyblivého obrázku. Tato kapitola má za úkol analyzovat vlastnosti videa vhodné pro použití v prezentaci. Otázek vizuálního vzhledu a estetiky se tato část dotýká pouze okrajově, protože účel a zaměření je směřováno více k technickým aspektům práce včetně problémů, které při manipulaci vznikají.

Množství informací obsažených v multimediálních datech jako video, zvuku apod. oproti textu je neměřitelný pojem, ale předpokládá se, že obraz má hustější informační hodnotu a video, jako sekvence obrazů, bude obecně uživateli poskytovat více informací než textová forma.

Společnost Eyetrack III provedla test [12], který nám dává částečnou odpověď na otázky ohledně srovnání textové a multimediální formy prezentace určité informace. Test se zaměřil především na schopnost lidí zapamatovat si a pochopit předložené informace. Shrnutí výsledků je následující:

- Pro správné vybavení si informací je pouze velmi malý rozdíl v tom, jestli byly získány z textu či videa.
- Vybavení si neznámého postupu či procedury je přesnější po shlédnutí multimediálního grafického formátu.
- Lidé si lépe vybaví faktické přesné informace z textové formy dat.
- Mezi muži a ženami je neznatelný rozdíl v ohledu na vybavení si informací prezentovaných v textové či multimediální formě.

Z tohoto testu je možné usuzovat, že ačkoliv je video stále populárnější [13], při volbě formy mezi prezentováním informace textově či videem, je vhodné přihlížet k obsahu a jeho účelu.

3.4.1 Video v 3D

Celkově se tato práce zaměřuje v otázce prezentací na pokročilé prvky jako je video a 3D. Aby byla analýza prostředků v tomto ohledu kompletní, je vhodné se zabývat i technologiemi, které mohou vzniknout kombinací základních prvků.

Video použité ve virtuálním prostoru je kombinace, která se v současné době příliš nepoužívá. V poslední době vzniklo např. několik webových prezentací:

Defqon 1 společnosti Hanazuki studio: http://events.q-dance.nl/2010-06_12-juni_defqon1/

The Economist: <http://thinkingspace.economist.com/>

Tyto internetové stránky kombinují použití videa a 3D, ovšem většinou nejde o rastrové video (pouze krátké sekvence), ale o animace, které vhodně propojují využití vektorových obrázků s definováním pohybu obrázků rastrových.

Tímto způsobem prezentovat informace je možné pomocí nástrojů VRML nebo Flash s využitím některého 3D enginu (nejčastěji Papervision). Problémy kompatibility tohoto řešení jsou popsány výše v kapitole 3.1.3 a při reálném použití existuje několik omezení.

V případě webové prezentace nebo obecně online prezentace je nutné zajistit, aby byl obsah dostupný, což znamená že objem stahovaných dat by neměl být příliš velký. Pro srovnání, běžná webová stránka bez obrázků má objem cca 100KB, s malými obrázky většinou více, obvykle do 500KB (např. idnes.cz), s rastrovým videem se objem webové prezentace pohybuje již v řádu jednotek až desítek MB.

Jako další omezení může být chápán samotný virtuální prostor, ve kterém je nejčastěji využíváno perspektivního zobrazení. Díky perspektivě jsou objekty v prostoru zkresleny, tedy video nanesené na těleso bude deformováno.



Obrázek 3.13: Nanesení videa na krychli a kouli ve VRML

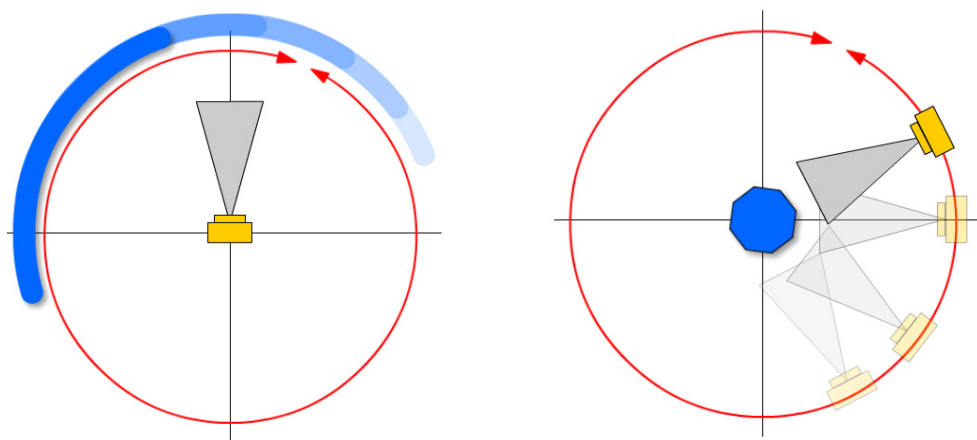
Video v prostoru není příliš vhodné pro jeho samotné sledování, v případech, kdy se prostorového videa využívá jde většinou pouze o vizuální doplňky bez důrazu na informační sdělení.

3.4.2 3D ve videu

Na rozdíl od technologie popsané v předešlé kapitole, je zobrazení prostoru ve videu výrazně častějším jevem. Výsledkem této kombinace prostředků je video, které je možné použít v prezentaci stejně jako jiná videa. Jde o sekvenci 2D obrazů, která nemusí obsahovat a také ve většině dnes používaných případech neobsahuje prostorovou informaci.

Jedním z nejsnazších způsobů jak prezentovat prostorový obsah, je vytvoření sekvence snímků nad rotovaným objektem. Sekvenci je možné automaticky přehrávat nebo, pro větší interaktivitu, je uživateli umožněno skrze uživatelské rozhraní spouštět sekvenci v libovolném směru a rychlosti. Objekt nebo scénu je možné prohlížet z různých směrů. Výsledné video pak může mít při vhodném nasnímání stejnou informační hodnotu jako původní prostorový objekt.

Snímky pořízené vždy s drobným otočením scény, nebo snímáče jsou vhodně seřazeny do sekvence, jak naznačuje obrázek 3.14. Nejlépe tak, aby byl vnímaný pohyb co nejplynulejší, zároveň by měly navazovat i začátek a konec sekvence pro případné přehrávání ve smyčce. Počet potřebných snímků je odvislý od rychlosti s jakou bude pohyb prezentován. Obecně není při rychlejším pohybu potřeba velkého množství detailů. Pro plynulou animaci je dostačujících 15 snímků za sekundu, naopak většinou není potřeba více než 25 snímků za sekundu, protože větší rozlišení již lidské oko není schopné vnímat.



Obrázek 3.14: Diagram znázorňující snímání prostoru rotací objektu (vlevo) a kamery (vpravo)

Interaktivní aplikace zobrazující prostor pomocí sekvence mohou obsahovat libovolné množství detailů. Většinou, nejde-li o video, je rychlost prohlížení volena uživatelem, takže čím více detailů má k dispozici, tím lepší si může udělat představu o prohlíženém prostoru či objektu.

Ve výjimečných případech, například máme-li malé množství snímků, je třeba zohlednit diskrétní počet snímků, kterých je konečný počet a neexistují mezi nimi přechody. Proto by rychlost zobrazování snímků a otáčení scény měla splňovat vzorkovací teorém:

$$v \geq 2B, \text{ kde } v \text{ je počet snímků za sekundu a } B \text{ počet otočení za sekundu.}$$

Nevýhodou takto sekvenčního prohlížení prostoru je omezení pohybu určené právě jen řadou snímků, které jsou předem dané a není možné je již interaktivně měnit. V mnohých případech je to však zcela dostačující řešení a i díky jednoduchosti pořízení, kdy není potřeba prostorová informace, kromě pořadí snímků, je tento způsob prezentace prostoru stále využíván (především pro prezentaci menších produktů jako jsou mobilní telefony, šperky apod.).

3.4.3 Analýza sledovanosti

Snaha analyzovat video, se zaměřením na sledovanost a s ní spojené informace, může být považována v oboru informatiky za okrajovou záležitost. Chováním lidí při sledování videa, případně dopady použitých informací a jejich formy, se více zabývají jiné obory jako sociologie, psychologie či ekonomie, zaměřené např. na televizní reklamu. Vše je však pevně spjato, neboť práce s informacemi, jejich uchovávání, zobrazování, analýzy a jiné činnosti, jsou lidé schopni provádět díky technice založené na informačních technologiích. Při viditelné potřebě propojení více oborů je pak výhodné, aby zúčastněné strany ovládaly i dovednosti přesahující jejich hlavní obor.

Analýza sledovanosti, která byla pro tuto práci provedena, je postavena na moderních informačních technologiích. Její účel či interpretace by se mohly zdát být za hranicemi použitelnosti pro informatiku. S ohledem na tvorbu prezentací ale přináší informace, které mohou reálně pomoci s vymezením některých důležitých aspektů výsledné prezentace.

Především je možné získat časový odhad délky prezentace, vhodné rozložení informací nebo časově-místní lokalizaci.

Jako nástroj pro sběr dat bylo využito internetového serveru youtube.com společnosti Google. Z dostupných zdrojů poskytuje nelepší možnosti sběru dat a jejich analýzy. Zároveň díky delšímu období sběru dat je statistický vzorek poměrně široký. Následující tabulka udává některé konkrétní sebrané hodnoty vázané na soukromý účet <http://www.youtube.com/user/mptasek>.

Období sběru dat	27.10.2007 až 27.4.2010
Počet nahraných videí	20
Nejkratší video	44 sec
Nejdelší video	9 min 31 sec
Průměrná délka videí	3 min 48 sec
Celkový počet shlédnutí	115 396

Tabulka 4: Statistické hodnoty vázané na účet serveru youtube

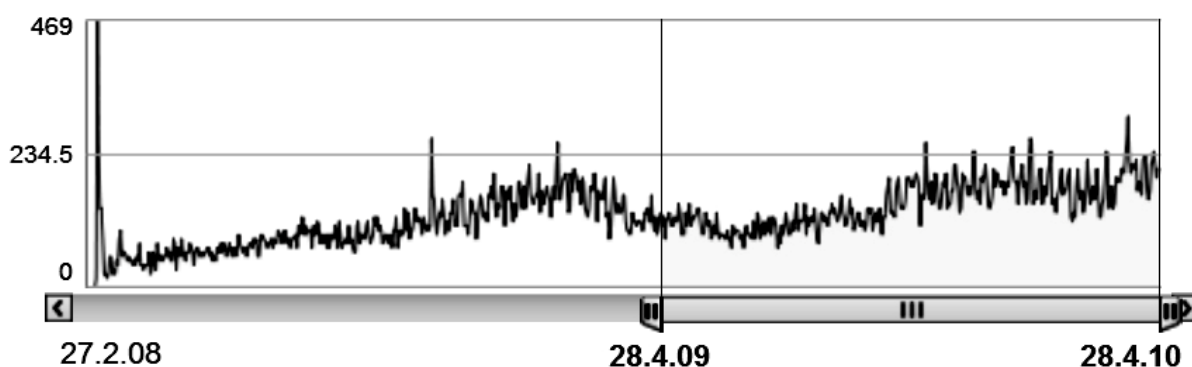
Interpretace získaných dat je věcí do jisté míry subjektivní, současně zůstává také faktem, že není možné objektivně kvantifikovat obsah jednotlivých sledovaných videí přesto, že obsah zásadní měrou ovlivňuje výslednou sledovanost.

Na základě získaných dat je možné vysledovat následující trendy, jak uživatelé přistupují ke sledování obsahu:

- Pozornost diváka je největší na začátku videa, přibližně do 45 sekund. Existuje velké množství uživatelů, kteří po této době ve videu přeskočí na závěrečnou část, kde pravděpodobně zjišťují jak výrazně se obsah v průběhu videa změnil. Tyto výsledky korespondují s obecnými doporučeními pro tvorbu videa a filmů jakéhokoli rozsahu. Nejdříve je nutné diváka přilákat a vynutit si jeho pozornost, je ale potřeba si zajímavé informace nechat i na konec, nejlépe s pointou, která celou prezentaci uzavírá.
- Výraznější obsahová změna závěru videa oproti jeho začátku vede k tomu, že lidé zhlédnou i prostřední část. Tento závěr vychází intuitivně z představy, že nezaznamená-li uživatel výraznější změnu mezi začátkem a koncem, předpokládá, že i střední část se nese ve stejném duchu. V opačném případě si střední část nelze domyslet, je tedy přirozené ji shlédnout či alespoň zkontrolovat.
- Největší pozornost diváka je možné si udržet u snímků trvajících přibližně do dvou minut. V tomto ohledu hodnoty příliš reflektují typ média, kdy na serveru nemohou být videa delší než 10 minut a uživatelé s tímto vědomím také ke sledování přistupují.

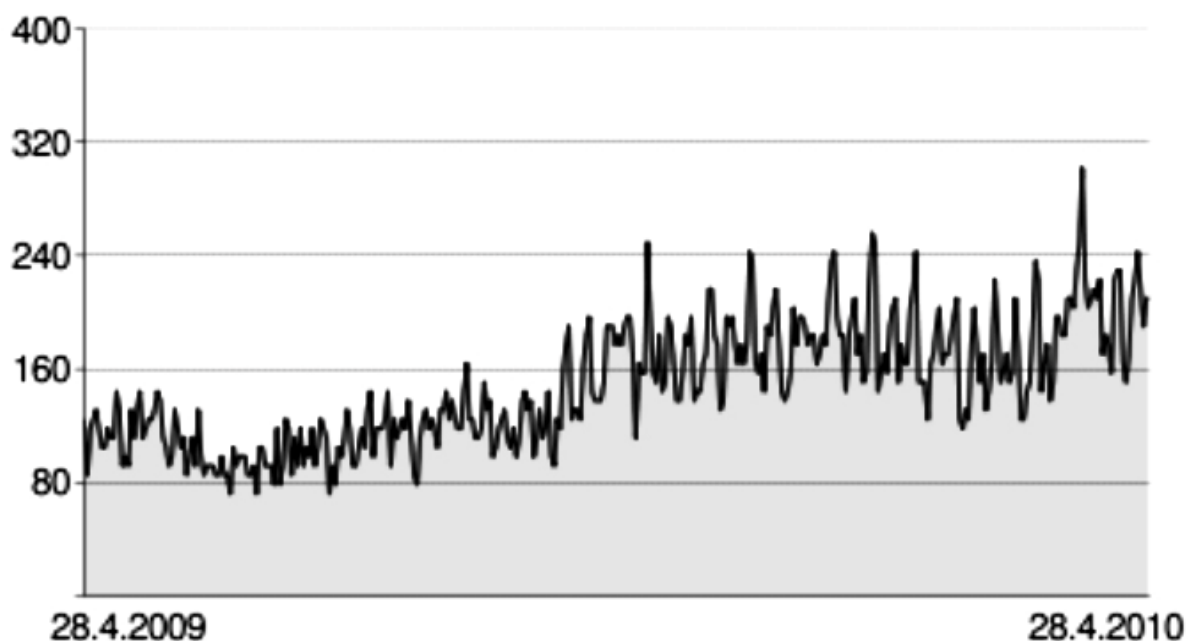
Autor práce připouští, že míra subjektivního pohledu a intuitivního přístupu (větší měrou vychází ze zkušenosti a obecných doporučení než matematického přístupu ke statistikám) v této kapitole může činit závěry pro někoho nepříliš oprávněné či objektivní. Skutečností ale zůstává, že obdobné statistiky jako poskytuje server youtube.com, jsou žádanou zpětnou vazbou pro samotné tvůrce obsahu a je možné obecně doporučit tvorbu prezentace společně s nástroji pro sběr statistických dat.

Následuje příklad obsahující grafy dat statisticky nejhodnotnějšího videa daného účtu. (převzato z webových stránek)

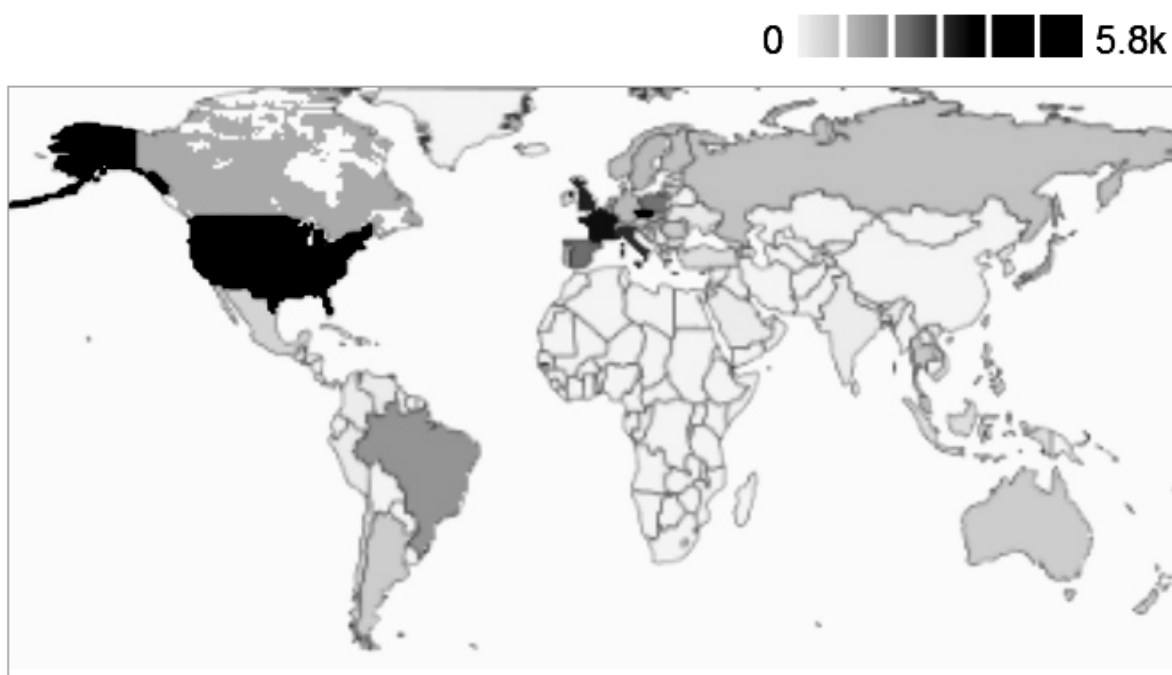


Obrázek 3.15: Graf sledovanosti (denní počet zhlédnutí) videa v celkovém období. Graf obsahuje možnost výběru specifického období pro podrobnější statistiku.

Na obrázku Obrázek 3.15 je vidět vývoj sledovanosti od 27.2.2008. V prvních dnech byl zaznamenán velký nárůst, který záhy opadl. Celkově ale sledovanost stoupá a v posledním měsíci se pohybuje přibližně kolem hodnoty 200 shlédnutí za den.

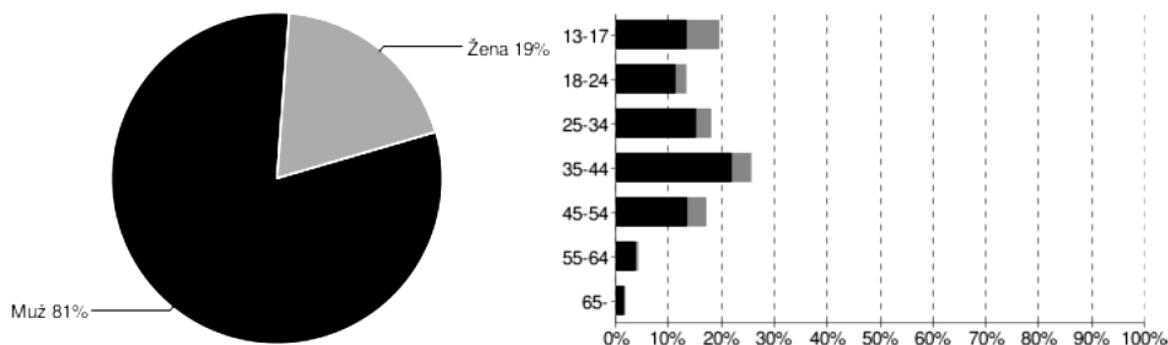


Obrázek 3.16: Graf sledovanosti, počet zhlédnutí za den ve zvoleném období (1 rok)



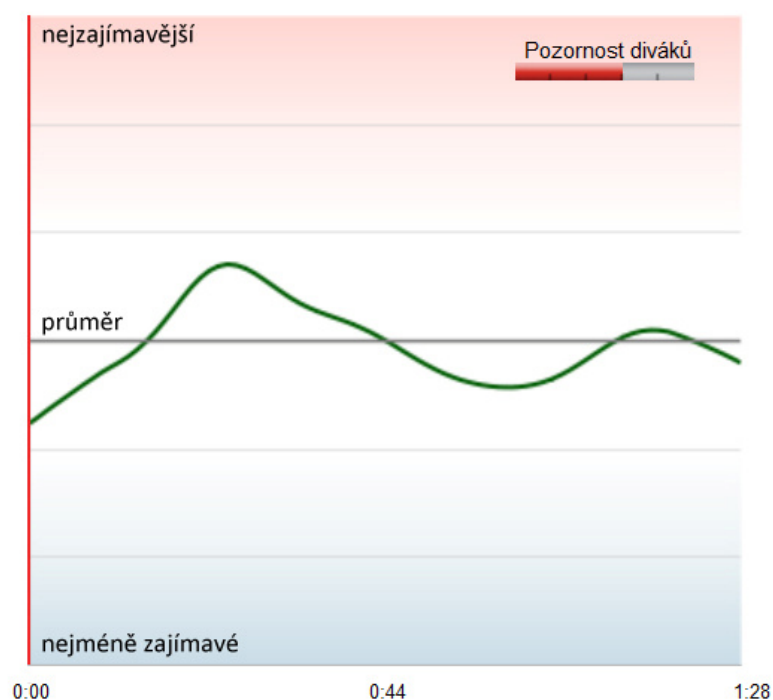
Obrázek 3.17: Počet zhlédnutí videa ve zvoleném období 28.4.2009 až 28.4.2010 podle zeměpisného umístění uživatelů. Tmavá barva představuje cca. 5800 zhlédnutí, světlá značí 0 zhlédnutí.

Z obrázku Obrázek 3.17 je možné vyčíst místní příslušnost diváků videa. V případě prezentace by tato informace mohla pomoci k lepšímu zaměření na určité země, které mají resp. nemají o prezentaci zájem např. volbou jazyka prezentace, titulků apod.



Obrázek 3.18: Demografické informace v daném období. Podíl mužů (81%) a žen (19%) při sledování videa (graf vlevo), věkové rozpětí diváků (graf vpravo) s naznačením podílu mužů (tmavě) a žen (světleji).

Podíl diváků na celkové sledovanosti podle jejich věku nebo pohlaví může být informací, která vypovídá mnohé o obsahu a jeho zaměření. Tvůrce videa si tak např. může snadno ověřit, zda se mu podařilo splnit původní zaměření na určitou cílovou skupinu lidí.



Obrázek 3.19: Graf sledovanosti v průběhu videa

Graf na obrázku Obrázek 3.19 znázorňuje statistiky pro jednotlivé momenty videa v porovnání s videy podobné délky. Čím vyšší je křivka grafu, tím oblíbenější video je: méně diváků video předčasně opouští a někteří mohou sledovat části videa opakovaně. Pozornost diváků je vhodným ukazatelem schopnosti videa udržet si obecnost.

3.5 Výběr z použitelných technologií

Analýza dostupných prostředků vede ke konkrétnímu výběru prostředků použitelných pro realizaci. Zároveň může být i vodítkem a obecným doporučením pro výběr daných technologií v návaznosti na potřeby tvorby vybraného typu prezentace.

Jako platforma prezentačního prostředí je zvolena technologie Flash, která je, i přes překotný vývoj Silverlight, stále obecně mnohem rozšířenější. Neměl by být proto problém se zobrazováním ve formě webové aplikace, ani aplikace desktopové.

Pro zobrazení 3D scén může být využito vhodného propojení jak samotných možností Flash, tak možností VRML a jeho zobrazení, protože obě technologie jsou realizovány jako zásuvné moduly webové stránky. Flash však bude potřeba rozšířit o engine Papervision, který se bude starat o zobrazování 3D obsahu ve formátu .dae. Plně interaktivní ovládání je poté potřeba realizovat vlastní implementací. V tomto případě půjde o realizaci pohybu ve scéně a dynamické ovládání prvků scény.

Výsledné 3D scény budou pro potřeby statické části prezentace nasnímány do výsledného videa s ohledem na to, aby výstup obsahoval reprezentativní část interaktivního prostředí.

Z důvodu zkoušky více možností přístupu, budou některé modely užité v 3D virtuální galerii nasnímány z reálných objektů pomocí 3D scanneru Konica Minolta Vivid 910 do formátu VRML 2.0. Další objekty, použitelné v prezentaci však mohou být také převedeny do formátu VRML. Veškeré modely tak budou přímo použitelné ve virtuální scéně prohlížené pomocí VRML prohlížeče.

Pro potřeby virtuální 3D galerie zobrazované pomocí Flash, bude potřeba výsledný formát konvertovat do formátu Collada. Prostředky potřebné k operaci převodu formátu VRML 2.0 či jiného formátu, užívaného editory prostorových modelů, na formát Collada byly analyzovány a testovány v kapitole 3.1.1. Výsledky budou shrnuty a diskutovány v kapitole zabývající se metodologií pro použití pokročilých prostředků v prezentaci.

4 Návrh a implementace

Součástí praktické stránky této práce je vytvoření demonstrační prezentace, která by měla obsahovat zkoumané pokročilé prvky videa a 3D. Z pohledu softwarového inženýrství je navrhovaná prezentace softwarovou aplikací, která existuje v rámci daného životního cyklu. Tato kapitola představuje fázi podrobného návrhu, na nějž navazuje vlastní realizace. Aby však mohla být prezentace úspěšně implementována, při jejím návrhu je potřeba zohlednit další aspekty tvorby prezentace. V předešlých kapitolách byla zkoumána celková situace a následně byly analyzovány možnosti dostupných technologií.

Vybrané technologie, i v případě že jsou vybrány vhodně, samotné k danému záměru nestačí. Prezentace, a to je třeba si při tvorbě uvědomit, je tvořena za účelem prezentovat určité informace lidem. V současné době se nedá říct, že by byly prezentace generovány automaticky strojově (v budoucnu se však tento fenomén může objevit), alespoň ve větší míře ne, a i v případě, že by tomu tak bylo, pravděpodobně by byla snaha vytvářet výstup tak aby připomínal produkt člověka.

Při tvorbě prezentace je tedy důležité vytvářet výstup vhodný pro lidi, myšleno tak, že je potřeba zohlednit jisté společné rysy diváků. Aspekty tohoto problému se zabývá např. psychologie reklamy, v tomto ohledu jde o téma mezioborové. S jistou mírou nadsázky lze říci, že dokonalou prezentaci nemůže vytvořit ani odborník psychologie ani odborník informatiky, ale člověk znalý obou oborů či tým lidí.

4.1 Zásady tvorby prezentace

Prezentace se dělí na několik druhů, nejčastěji podle způsobu užití prostředků a jejich výběru. Zásady tvorby prezentací se v mnoha bodech shodují pro všechny jejich druhy, v čemž se shodují téměř všichni autoři prací v tomto oboru [14-16]. Přesto bude pro tuto práci účelné omezit se pouze na zásady, které se vztahují k prezentacím jež jsou předmětem této zprávy, tedy multimediální aplikace přehrávané na počítači nebo podobném zařízení.

Kroky procesu tvorby prezentace:

1. Definovat účel - většinou se jedná o poskytnutí informací nebo podpoření rozhodovacího procesu diváka.
2. Vymezit publikum - určit počet lidí, kteří pravděpodobně prezentaci zhlédnou, zaměřit se na určitou skupinu (např. ženy, muže, věkovou skupinu, podle odbornosti). Obecně, čím větší počet parametrů určíme, tím přesněji je možné prezentaci zacílit.
3. Zvolit prezentační metodu - i když se omezíme pouze na multimediální prezentace jak bylo uvedeno výše, stále existují další rozlišení, kterými je možné se zabývat. Např. zda je

prezentace na internetu / přístupná veřejně nebo pouze pro určitou skupinu uživatelů / přístupná interaktivně nebo s určitým omezením (pouze obrazovka) apod.

4. Definovat obsah - tvorba obsahové části je jedna z nejdůležitějších činností, která navíc velkou měrou určuje i následující kroky.
5. Vytvořit grafický návrh - tento krok je výhodné promýšlet v kontextu předchozích kroků.
6. Zvolit prostředky - vybrat vhodné technologie tak, aby efektivně předaly obsah v požadované formě. Z technického hlediska je nezbytné pro prezentaci zajistit vhodné technické zázemí. To souvisí i s výše uvedenými kroky. Problémy, které mohou vzniknout jsou např. u webových stránek jejich nedostupnost, neaktuálnost, v případě projekcí prezentací obrazová kvalita apod.
7. Vytvořit prezentaci - implementovat prezentaci na základě předchozích kroků.
8. Prezentovat - výsledek je nutné předat divákovi tzn. podle předchozích zvolených kroků např. umístit na internet či jiná distribuční média.

Výše uvedené kroky představují fáze, které je vhodné při tvorbě prezentace vytvořit. Současně však jsou některé důležitější a jiné méně, což koresponduje i s úsilím, potřebným k jejich zpracování.

Následuje stručné shrnutí obecných zásad i konkrétních doporučení, jaké vlastnosti by prezentace měla mít:

- Silný první dojem - první dojem (kladný nebo záporný) si divák vytvoří během prvních 7-10 vteřin. Je tedy vhodné v této době zaujmout např. originálním grafickým zpracováním nebo obrázkem, výstižným titulkem apod.
- Získat si diváka - v průběhu prvních 60 vteřin by měl divák dostat jednoznačné informace, které mu řeknou co je předmětem a účelem prezentace. Po uplynutí doby přibližně 2 minut zájem nerozhodnutého diváka výrazně opadá.
- Působit nadšeně - člověka je obecně snazší přesvědčit mírou nadšení, než vědomostmi. V případě multimediální prezentace to znamená podávat informace se zaujetím (k tomu může dopomoci živý vizuální styl). Doporučuje se kombinovat prvky zábavné a informativní přibližně ve stejném poměru.
- Výstižný obsah - v rámci podmínek množství informací, které má prezentace předat, je vhodné ponechat obsah stručný a výstižný. Znamená to případně volit mezi informacemi, které nejsou nezbytné, ty, jejichž uvedení nebude diváka rozptylovat či odvádět od hlavního tématu. Udává se, že ucelený blok prezentace by měl veškeré potřebné informace předat nejvýše v rozmezí 5-10 minut.

4.2 Téma a obsah prezentace

Návrh tématu a obsahu by měl vycházet z předchozích analýz kapitoly 3, společně s návazností na kapitolu 4.1.

Účelem prezentace je vhodně demonstrovat možnosti, které poskytují dnešní nástroje pro tvorbu prezentací s pokročilými prvky videa a 3D, což vychází ze zadání této práce.

Předpokladem je, že publikum bude převážně z řad studentů se zájmem o informační technologie, přičemž ale jejich znalost není vyžadována.

Prezentace je určena pro veřejnost, měla by být přístupná na internetu, zároveň však bude snaha zachovat vlastnosti aplikace takové, aby byla použitelná i bez internetového připojení.

Multimediální prezentace bude tvořena dvěma hlavními částmi. Částí interaktivní prezentace s prvky 3D a částí prezentace statické, která se zaměří na použití videa. Každá z nich pak vychází z řešení specifických problémů, kterými se zabývá tato technická zpráva.

Hlavním pilířem statické verze je výsledné propagační video, které bude obsahovat krátký příběh propojený obsahově s Ústavem počítačové grafiky a multimédií. Statická verze neumožňuje zaujmout diváka interaktivními prostředky, proto bude obsahovat více rozličných informací, které budou zobrazovány dynamickým stylem. To by mělo ve výsledku diváka zaujmout a udržet jeho pozornost. Zároveň by video mělo působit na subjektivní vizuální cítění diváka takovým způsobem, aby byly dodrženy zásady dobré prezentace. Předpokladem je použití filmových výrazových technik doplněných o vlastní speciální digitální efekty.

Interaktivní část by měla mít obsahově podobnou náplň jako statická část. Navíc však bude obsahovat právě interaktivní prvky. Struktura bude taková, že hlavní rozcestník aplikace povede do jednotlivých částí prezentujících určené informace. Obsahem bude jak část videa, která bude převzata z části statické, tak doplňující textové informace s hypertextovými odkazy. Pro demonstraci možností pokročilých prvků bude součástí prezentace také interaktivní virtuální 3D galerie, obsahující možnost prohlídky fakultního areálu a vhodné objekty, které obsahově souvisí s ústavem grafiky a multimédií.

4.3 Návrh struktury prezentace

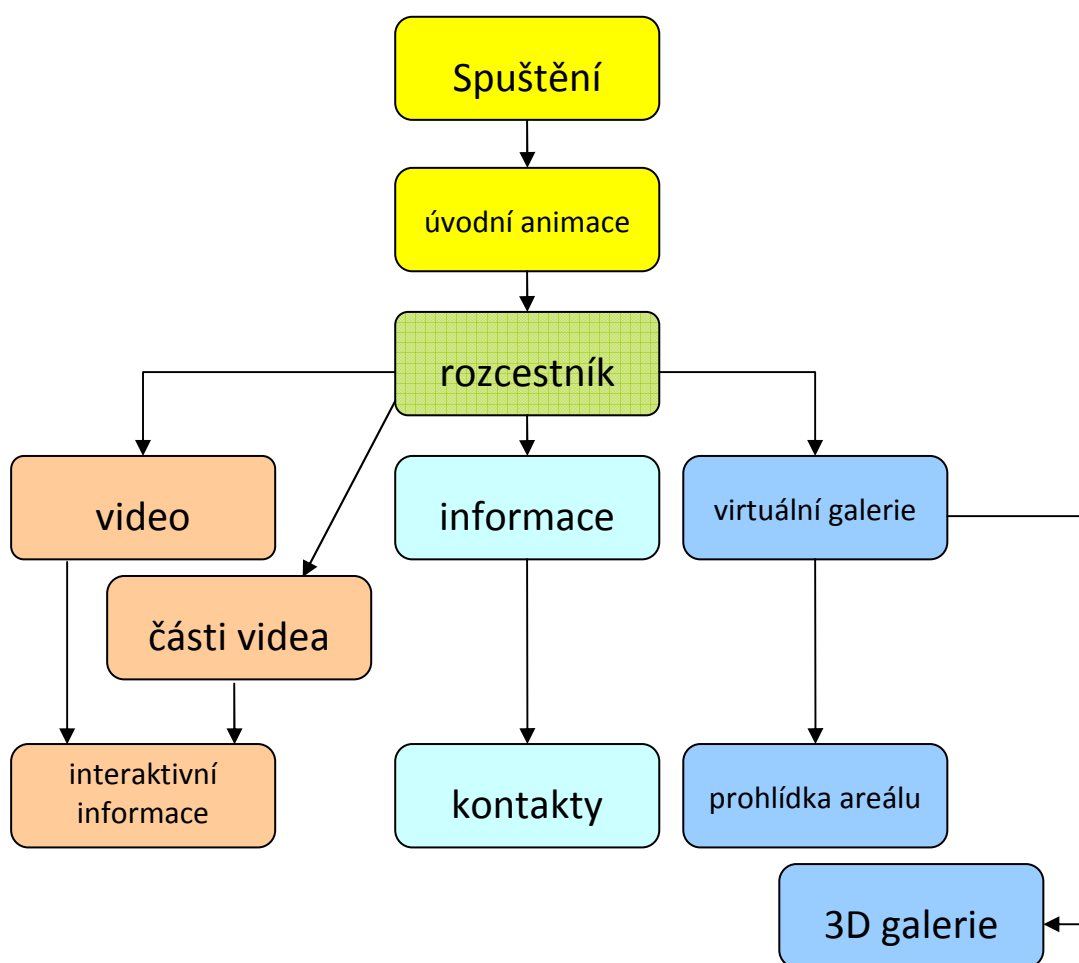
Návrh celkové struktury je založen na předchozím návrhu obsahu prezentace, zároveň jde o důležitou část definující směr a omezení následujících kroků návrhu.

Záměrem je, aby prezentace byla součástí jednotného grafického rámce bez rozlišení či vymezení speciálního místa pro uživatelské rozhraní. Předpokladem je, že ovládání aplikace bude jednoduché tak, aby mohly být ovládacími prvky pouze tlačítka přepínající na zvolený obsah.

Struktura aplikace je navržena způsobem, jaký je naznačen na následujícím obrázku Obrázek 4.1. Součástí obsahu mají být multimediální prvky. Protože se však má prezentace objevit i na internetu, u kterého se předpokládá přenosová rychlost dat nižší než u jiných řešení (DVD,

desktopová aplikace), je nutné strukturu aplikace navrhnout tak, aby prvky jako video či 3D scény byly od hlavní části odděleny. Načítáním jednotlivě, až v případě zájmu uživatele o daný obsah se zajistí co možná nejkratší reakce aplikace. Zároveň tak vzniká prostor během doby načítání objemnějšího obsahu, kdy je potřeba uživateli podat informaci o tom, že je aplikace aktivní, případně jaká je předpokládaná doba načtení.

Rozdílná povaha zobrazovaných dat a nutnost koordinace jejich načítání z externího zdroje bude součástí rozcestníku, který tak tvoří jádro celé aplikace. Pro prohlížení a ovládání jednotlivých částí bude vytvořena sada potřebných ovladačů.



Obrázek 4.1: Struktura interaktivní části

Obsahová struktura pomáhá vymezit návrh vizuálního funkčního rozložení jednotlivých oddílů. V tomto ohledu byla snaha využít inovativních postupů na základě analýzy prostředků, konkrétně bylo navrženo umístit základní část aplikace (rozcestník) do 3D prostoru a současně se tak pokusit vhodně propojit uživatelské rozhraní prostorové i ploché. V případě videa, virtuální galerie a části informací, je nutné použít různé ovládací prvky. Z toho důvodu byl design aplikace navržen ve třech vrstvách, které se v pomyslném prostoru překrývají.

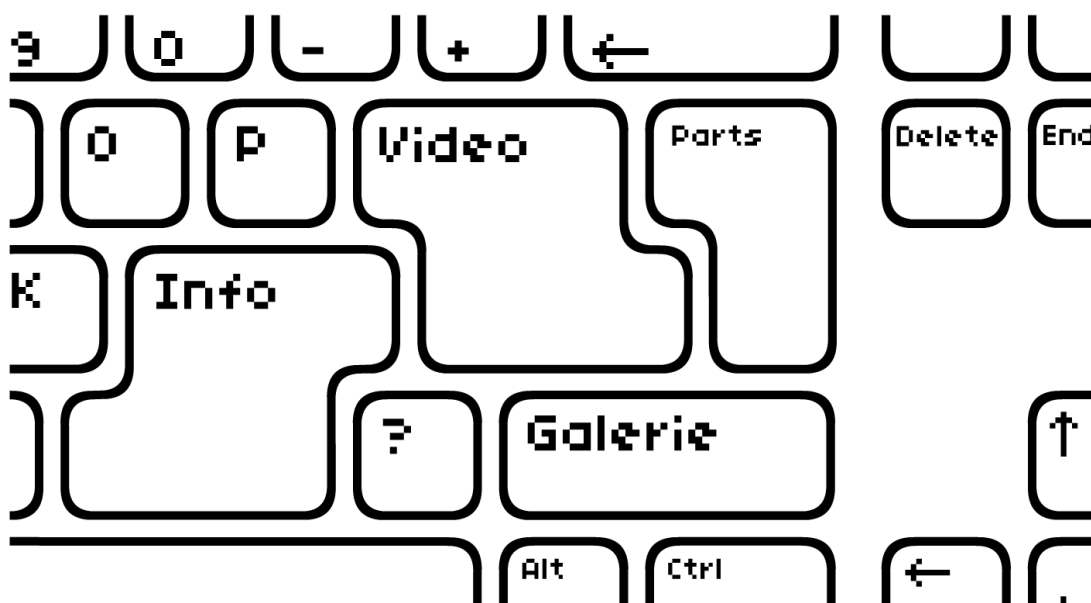
zůstane vizuální styl čistý. Ovládací prvky v bílé barvě budou vůči barvě pozadí dostatečně výrazné, z toho důvodu je možné je ponechat bez výplně (průhledné), tvořené pouze bílou konturou.

Rozlišení prezentace je určeno na 1024 x 576 pixelů. Tento rozměr je volen záměrně s ohledem na použití ve webové stránce, kde se rozměr šířky obsahu 1024 pixelů stále považuje za standardní hodnotu. Rozměr výšky 576 pixelů je dopočtem do poměru stran 16:9 tak, aby použité video mohlo být zobrazeno bez případných okrajů přes celou plochu prezentace. 3D modely budou zobrazovány na místě obsahové karty. Předpokladem je, že se v jednu chvíli bude zobrazovat pouze jeden vybraný model. Je to z toho důvodu, že modely jsou složeny z prostorových dat a textur, které je nutné načíst, zároveň však jsou složitější scény pro počítač značnou zátěží, což způsobuje zpomalení chodu aplikace.

Průchod prezentací je navržen tak, že si uživatel vybírá požadovaný obsah příslušnými tlačítky, která jsou graficky odlišena. Návrat na předchozí obsah je možný pouze tlačítkem zpět v pravém horním rohu aplikace, které má vždy stejnou funkci.

Menu rozcestníku musí obsahovat minimálně čtyři tlačítka, která budou ovládat přepínání prezentace do zvolených sekcí. Tlačítka budou umístěna na ploše střední vrstvy. Nápad vytvořit menu jako výřez zjednodušené klávesnice se inspiroval jedním z nejpoužívanějších a zároveň nejstarších periferních zařízení počítače. Tento pocit by měl podpořit i zvolený font písma podobný písmu starších počítačů. Jde o autorský návrh, který se snaží využít toho, že klávesnice je každému uživateli počítače natolik známá, že i jen pouhé rozmístění a pozice několika tvarů by mělo stačit k tomu, aby evokovaly právě klávesnici. Mimo funkční klávesy však bude menu obsahovat i ostatní klávesy výřezu klávesnice, společně s jejich obvyklými popiskami, což záměr ještě podpoří.

Grafická podoba jednotlivých tlačítek je inspirována tlačítky klávesnic používaných na dotykových displejích moderních zařízení. Tvar vychází z čtverce se zaoblenými rohy jak je vidět na následujícím obrázku.



Obrázek 4.3: Grafická podoba hlavního menu rozcestníku

Většina tlačítek standardní klávesnice je určena pro právě jeden znak, text označující hlavní tlačítka menu (Video, Galerie, Info) by musel být na jednom tlačítku zmenšen či zalomen na více řádků, tím by se snížila celková přehlednost. Proto byla sloučena čtyři tlačítka v jedno, což vzhled mírně zjednodušilo.

Výsledné rozmístění a tvar tlačítek je podobné některým typickým tvarům počítačové hry Tetris. Tato podobnost byla využita a dále rozvedena animací při nájezdu menu. Tlačítka se objevují jednotlivě v horní části a typickým pohybem (jako ve hře Tetris) sestaví konečnou podobu menu.

4.5 Návrh programové části

V této kapitole jsou navrženy konkrétní prostředky a způsob, jakým budou implementovány. Návrh je dekompozicí celkového problému na jeho jednotlivé podčásti.

4.5.1 Datová část

Veškerý obsah prezentace je zastoupen textovými informacemi, videem, protorovými modely a grafickým uživatelským prostředím. Grafické uživatelské rozhraní je navrženo poměrně jednoduše, k jeho realizaci se nabízí využití vektorové grafiky, kterou je možné vytvořit přímo v prostředí Flash editoru. Vektorová grafika má výhodu menšího objemu dat a není vázána na určité rozlišení.

Textové informace je možné použít dvojím způsobem. V případě samostatné aplikace bez připojení k internetu je nutné text nahrát do knihovny Flash editoru a umístit do prezentace společně s grafickými prvky uživatelského rozhraní. V případě online aplikace umístěné na internetu je možné využít odkazu na webovou stránku.

Video používané v prezentaci by mělo mít rozměr 1024x576 pixelů, bude umístěno externě v adresáři, který bude připojen k prezentaci. Je nutné, aby video bylo ve formátu .flv případně .swf. Video snímky budou produktem fáze vizualizace 3D modelů pomocí programu CamStudio společně s autorskou vizualizací animace 3D scény vytvořené v programu pro 3D modelování.

Prostorové modely budou produktem scanování na 3D scanneru Konica Minolta Vivid 910, společně s modely použitými ve vizualizaci 3D scény a modelem areálu FIT VUT. Pro zobrazení vybraných modelů ve vlastním okně aplikace je potřeba využít technologie Papervision. Modely budou načítány podobně jako video, z externího adresáře a musí být ve formátu .dae. V tomto směru by bylo vhodné vytvořit automatizovaný prostředek, který bude modely před použitím v prezentaci převádět na požadovaný formát .dae. Současně je potřeba zajistit aby rozměry modelů byly v zobrazitelných mezích.

Pro zobrazení modelů technologií VRML je nutné v prezentaci použít odkaz na externě uložený model ve formátu .wrl. Zároveň je nutné, aby byl na daném počítači nainstalován zásuvný modul internetového prohlížeče pro zobrazení VRML, nebo jiná aplikace pro zobrazování tohoto formátu.

4.5.2 Řídicí část

Řízení a zobrazování větší části prezentace zajišťuje technologie Flash. Pro demonstraci širších možností zobrazení prostorových modelů, bude však využito vedle Flash i technologie VRML.

V případě VRML je situace poměrně jednoduchá, neboť prohlížeč VRML (doporučený je Cortona3D) je možné otevřít odkazem na vybraný soubor .wrl. Asociace operačního systému s daným formátem zajistí otevření příslušného programu, ve kterém je již veškeré uživatelské rozhraní zabudováno.

Většina jednodušších aplikací technologie Flash je kompilována společně s obsahem. To má na jedné straně své výhody, např. jediný soubor obsahuje veškerá data, na straně druhé jsou zde však nevýhody v podobě většího datového objemu a také nemožnosti měnit vlastnosti aplikace bez nutnosti kompilace zdrojového kódu. Případné přizpůsobení, které musí být implicitně součástí, je z programového hlediska náročnější variantou (kvůli složitosti i udržitelnosti) než použití externího souboru s konfiguračními daty.

Za účelem uložení a snadného načítání konfiguračních dat je s ohledem na Flash výhodné použít souborů XML. Pro snadnou manipulaci s tímto formátem dat je Flash vybaven několika knihovnami a navržená řídicí část bude postavena právě na této variantě.

Aplikace je navržena tak, aby obsahovala možnost prezentace videa i 3D a použití dalšího softwarového vybavení nebylo nutné. V nejširším pohledu funguje prezentace ve třech krocích:

- Úvodní část obsahuje animaci, která představuje uživateli téma prezentace. Animace musí být datově málo objemná, protože bude zároveň plnit funkci tzv. preloaderu, tedy místa, ve kterém se načítá zbylá část aplikace. Během načítání bude zobrazena i informace o jeho průběhu.
- Menu rozcestníku tvoří grafická animace následovaná částí s možností výběru příslušného obsahu. Během krátké animace je vhodné, aby se na pozadí nahrály řídicí proměnné, určující především adresy k externě uloženým souborům. To zaručí, aby při chybném načtení menu reagovalo případným odebráním funkce tlačítka, které v dané chvíli neodkazuje na relevantní data.
- Výběr příslušného obsahu probíhá stisknutím tlačítka myši. Zároveň, protože je menu graficky navrženo jako klávesnice, by měl fungovat i výběr tlačítka pomocí klávesnice. Stavové řízení bude zajišťovat zobrazení správného obsahu společně s potřebnou změnou uživatelského rozhraní.

4.5.3 Uživatelská rozhraní

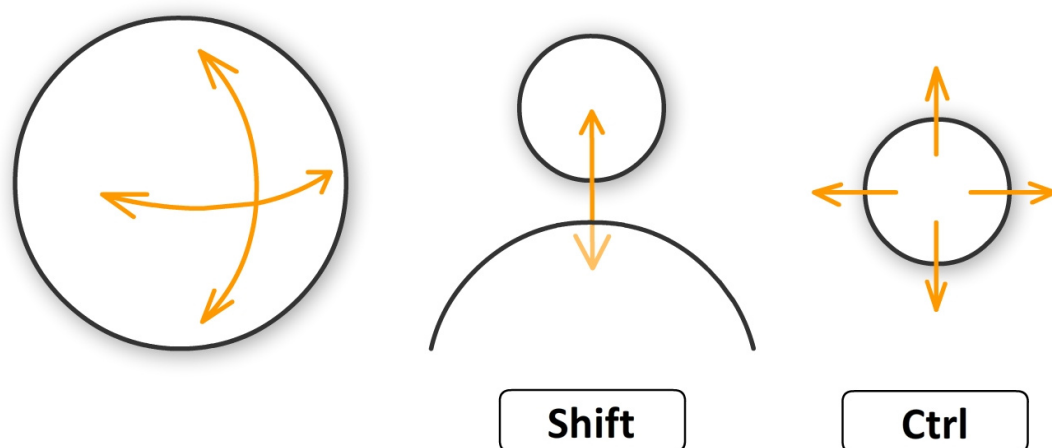
Hlavním ovládacím zařízením prezentace je myš. Význam klávesnice je spíše doplňkový.

Uživatelská rozhraní aplikace je možné rozdělit do čtyř typů:

Menu rozcestníku je tvořeno graficky výraznými tlačítky. Při nájezdu myši na vybrané tlačítko, se změní jeho barevná výplň. V tu chvíli bude výrazně odlišeno od ostatních tlačítek a uživatel jasně uvidí, které si zvolil. Při aktivaci výběru je uživatelsky přívětivé, aby byla jeho akce potvrzena a obsah se příslušně změnil. Toho je možné docílit tak, že prostorová plocha např. na chvíli změní svou pozici či otočení. Aby však bylo patrné, že jde o plochu umístěnou v prostoru, je mírně natočena do místa kurzoru myši, což by mělo vypadat dynamicky a díky perspektivě bude přivrácená strana rozměrově větší.

Rozhraní videa a jeho částí je tvořeno z uživatelského pohledu tlačítky pro zastavení videa a opětovné přehrání, ovladačem hlasitosti zvuku, případně přetočením na následující snímek. Z programového hlediska je nutné implementovat funkci tlačítek, nahrání příslušných dat a jejich obsluhu. S ohledem na možnosti, které technologie Flash poskytuje je vhodné implementovat rozšíření videa o možnost vkládat titulky či jiné textové informace bez nutnosti kompilace.

Rozhraní 3D části musí zajistit vhodné ovládání pro prohlížení prostorového obsahu. Kompletní možnost prohlížení je kombinací možnosti otáčení modelu, přiblížení a posunu. Flash, ani nástavba pro práci s 3D, implicitně uživatelské rozhraní neimplementuje. Otáčení je vhodné realizovat změnou natočení objektu v prostoru. Vazba na objekt kurzoru myši vymezuje možnost rotace pouze ve dvou osách, což je ale pro prohlížení dostatečné řešení, neboť objekt bude viditelný ze všech směrů. Ovládání je navrženo jako kombinace pozice kurzoru myši s levým tlačítkem a klávesami "Shift" a "Ctrl". Zároveň s uživatelským rozhraním je potřeba zajistit i načtení a zobrazení vybraného obsahu.



Obrázek 4.4: Naznačení manipulace s prostorovým objektem

Výběr z kolekce je součástí ostatních menu, která nebyla zmíněna výše. Na rozdíl od menu rozcestníku, kde je počet ovládacích prvků neměnný, může sekce galerie obsahovat dynamicky se měnící počet položek na základě konfiguračního souboru. Kvůli možnosti snadného výběru uživatelem je vhodné zobrazit tlačítka s ohledem na obsah společně se stručným textovým popisem.

4.5.4 Konfigurační XML

Použití XML je výhodné z několika důvodů. Jedním z nich je jednoduchá a udržitelná struktura, zároveň je možné data opatřit vlastními atributy. Další výhodou je široké rozšíření podpory pro manipulaci s formátem XML.

Z předchozího návrhu aplikace vyplývá, že je potřeba navrhnout strukturu celkem tří XML souborů. Jejich tvar je zásadní i pro případné dynamické generování serverem nebo skriptem při použití na internetu. Návrh XML schématu je vytvořen pomocí jazyka XML Schema, který je přijat a doporučován k tomuto účelu konsorciem W3C.

XML pro načítání obsahu

V návrhu je potřeba vyřešit možnost použití různých datových typů, které definují obsah nahrávaný do prezentace. S ohledem na možnost širšího uplatnění modulu aplikace i v jiných realizacích, obsahuje schéma různé typy multimediálních dat, které se však v dané aplikaci nemusí nutně objevit. Dále je obsažena možnost strukturování obsahu do složek, což je výhodné v případě, kdy prezentace obsahuje velké množství souborů, např. obrázkovou galerii. Možnosti aplikace se tímto řešením značně rozšíří a obsah prezentace bude svázán s konkrétní realizací menší měrou.

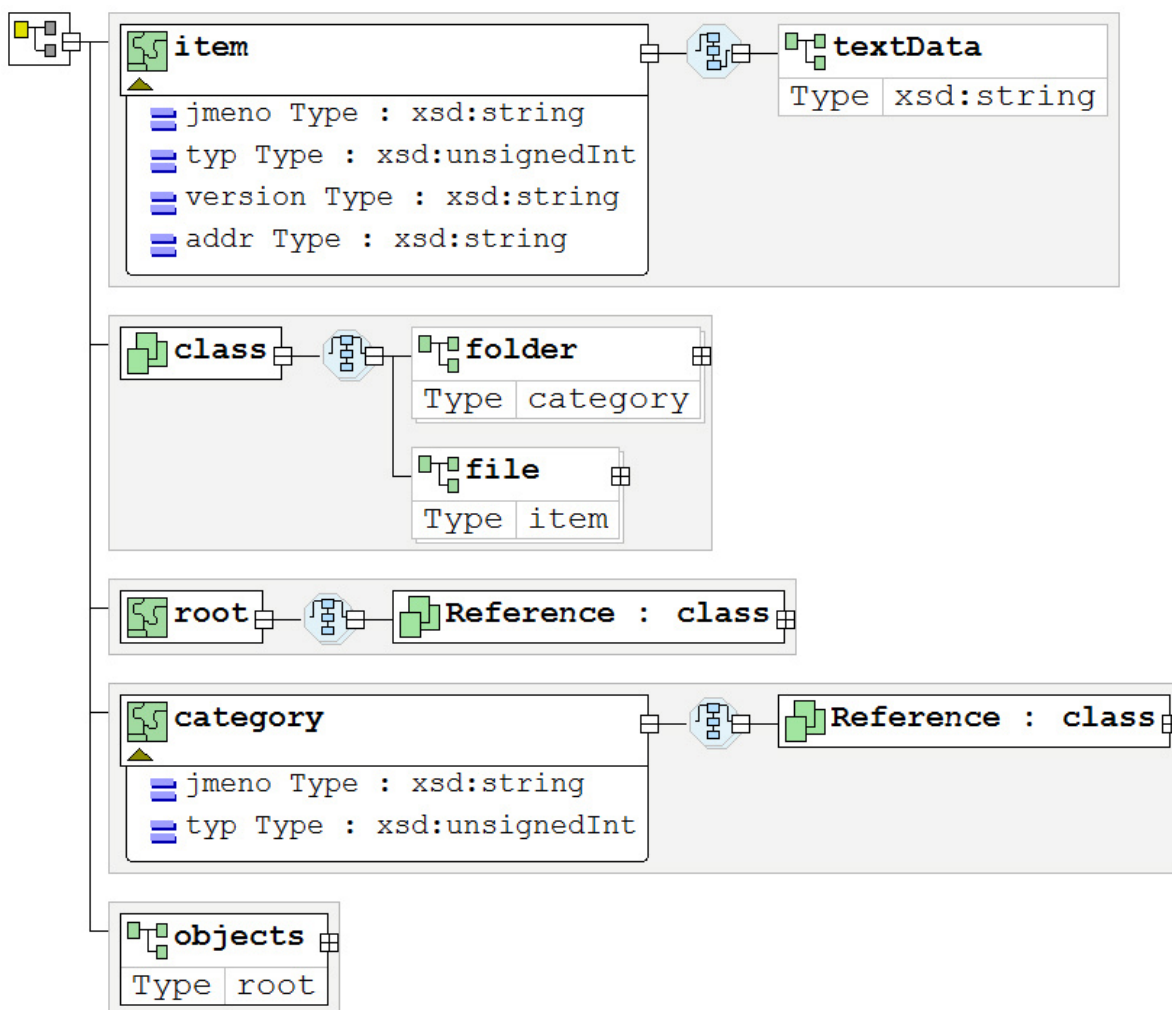
Součástí adresáře je nejen jeho název, ale i typ určující kategorii dat, která jsou v adresáři obsažena (to může případně sloužit ke grafickému rozlišení adresářů). Shodný typ je používán i pro jednotlivé položky, obecně se však typ adresáře a položek v něm umístěných může lišit.

Kategorie dat	číselná hodnota	formáty
text	0	obyč. text; html
obrázky	1	JPG, PNG, GIF
zvuk	2	MP3
3D	3	DAE
video	4	FLV; SWF
ostatní	5	

Tabulka 5: Přiřazení číselné hodnoty dané kategorii datového obsahu

Na následujícím obrázku 4.5 je znázorněn konkrétní návrh. Vytvořené XML schéma, které je jeho předlohou, je součástí přílohy A.3.1. Na jeho struktuře je vidět, že možnost zanoření adresáře do jiného adresáře je realizována vytvořením nových datových typů s referencí.

Součástí jednotlivých položek může být textová informace, takže není potřeba načítat další soubor, Flash navíc umožňuje zpracování textu s HTML značkami. Volitelným parametrem je adresa k souboru a v případě 3D formátu i jeho verze. Jednotlivé formáty, které je možné zpracovat, jsou popsány v kapitole 4.5.1. Z formátů obrázků je možné použít JPG, PNG a GIF. Zvuk lze načíst ve formátu MP3.



Obrázek 4.5: Diagram XML schématu pro načítání obsahu

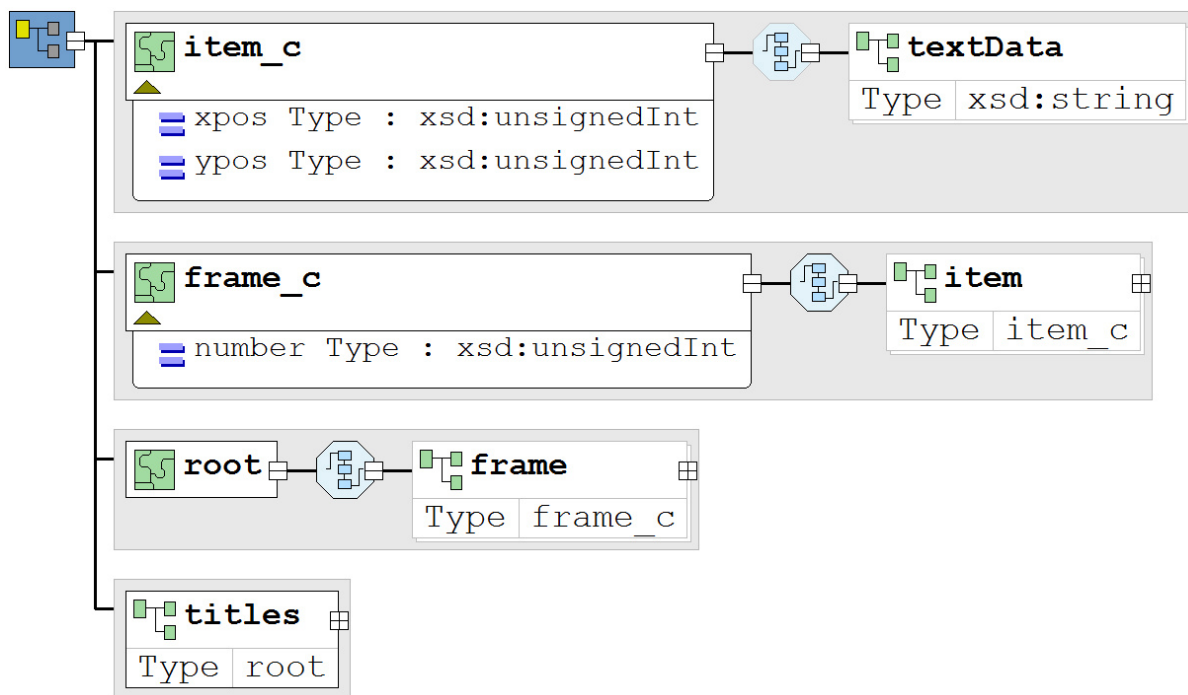
XML pro generování textu ve videu

Technologie Flash funguje po takzvaných snímcích, kde každý snímek je možné si v animaci představit jedním obrázkem. Na každý snímek je možné připojit libovolný program a při přehrávání videa je přístupná informace, na kterém snímku videa se aplikace nachází. Z tohoto předpokladu vychází schéma XML pro generování textu.

Po načtení příslušného konfiguračního XML je pro zobrazení textu nutné znát čas (číslo snímku od začátku), pozici, kam informaci na ploše umístit, a zobrazený text, který je možné formátovat pomocí HTML. Schéma je navrženo tak, že je možné v každém snímku použít neomezený počet textových informací, každou na příslušné pozici.

Modul bude ovládán stavovým řízením tak, že nastavená hodnota bude vždy používána až do její změny. Aplikace musí navíc obsahovat prvotní nastavení pro případ, že např. pozice titulku není určena.

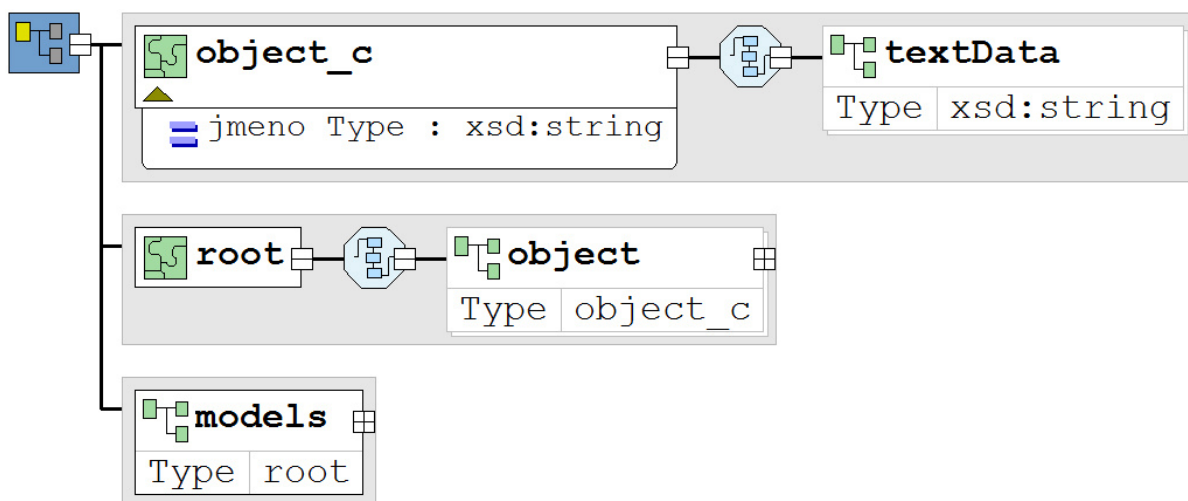
Na následujícím obrázku 4.6 je diagram XML schématu pro generování textu ve videu, skutečná podoba schématu je v příloze A.3.2.



Obrázek 4.6: Diagram XML schématu pro generování textu ve videu

XML pro skript předzpracování

Návrh počítá s realizací skriptu pro předzpracování 3D objektů. Pro potřeby skriptu je navržena struktura XML souboru, jak je naznačeno v následujícím diagramu. Schéma na obrázku 4.7 je poměrně jednoduché, protože funkce konverze převádí daný soubor na soubor stejného jména pouze v potřebném formátu s příponou .dae. Skutečné schéma je umístěno v příloze A.3.3.



Obrázek 4.7: Diagram XML schématu pro skript předzpracování

4.5.5 Předzpracování 3D objektů

Prostorové objekty použité v prezentaci mohou obecně pocházet z různých zdrojů (3D scanner, vlastní model, cizí model, aj.). Z toho vyplývá, že není možné se automaticky spolehnout na výsledný rozměr objektu, které mohou být tvořeny v různých souřadných systémech různých jednotek.

Dalším parametrem, který určuje velikost vykresleného modelu v prezentaci je prostorová vzdálenost od kamery, tedy od plochy zobrazované uživateli.

Řešení problému je navrženo tak, že bude vytvořen prostředek, který normalizuje velikost všech objektů na potřebnou velikost a zároveň realizuje převod formátu na výstup použitelný v aplikaci založené na technologii Flash. Návrh je kompromisem mezi složitostí implementace prostředku a možností provádět úpravy manuálně.

Aplikace Blender určená pro práci s 3D obsahem je pro problém předzpracování téměř ideální. Blender je možné spouštět z příkazové řádky bez nutnosti zobrazení grafického uživatelského rozhraní, což celý proces výrazně urychlí. Po provedení operací určených příslušným skriptem se aplikace opět ukončí.

Skript je nutné vytvořit v jazyku Python s využitím vnitřních funkcí Blenderu. Modely, jejich adresy a jména budou skriptem načítány z připraveného XML, jehož schéma bylo navrženo výše.

4.6 Implementace

V této kapitole bude detailně popsán způsob implementace každé z částí výsledné aplikace. Jak vyplývá z předchozí kapitoly návrhu realizace, v konečném výsledku je propojeno několik různých technologií. Hlavní část však představuje prostředek Flash využívající k implementaci řízení ActionScript 3.0.

Další specifickou vlastností Flash je využití objektů MovieClip, které mohou obsahovat další zanořené entity (jedná se o jeden z vůbec nepoužívanějších objektů). MovieClip navíc může existovat jako grafická struktura obsahující libovolný obsah, nejčastěji animace, obrázky aj. Je možné s ním pracovat i virtuálně a využít velký počet atributů a metod, které se k tomuto objektu váží.

4.6.1 Preloader

Jak napovídá název této části, preloader je modul umístěný před načtením hlavního obsahu prezentace. Slouží k informování uživatele o průběhu načítání, a protože technologie Flash pracuje na základě snímků časové osy, preloader je umístěn na prvním snímku. Po načtení příslušných dat je na 6 sekund zobrazen informační panel, následně je zobrazena hlavní část aplikace.



Obrázek 4.8: Grafické znázornění průběhu načítání dat

Implementace preloaderu spočívá ve vytvoření objektu posluchače událostí (EventListener). Při změně události progress je volána funkce `update()`, která přepočítává aktuální stav načtených dat na procenta podle následujícího vzorce: $percent = (bytesLoaded * 100) / bytesTotal$.

Výsledná hodnota je zaokrouhlena, aby mohla být použita jako číslo snímku, na který se animace průběhu načítání posune. V případě načtení všech dat se řízení aplikace přesune na následující snímek.

4.6.2 Hlavní část aplikace

Hlavní část aplikace představuje pomyslné jádro celé aplikace, které slouží k zobrazení prezentace, jejímu řízení a obsluze. Ve Flashi je možné umístit zdrojový kód přímo na vybraný snímek časové osy, ovšem výhodnější se zdá být možnost umístit zdrojový kód do externího souboru s příponou `.as` (actionScript). Takto je možné soubor prohlížet a editovat v libovolném textovém editoru.

Soubor `Main.as` vytváří vlastní třídu objektů, která je do hlavní zdrojové části importována. Vytvořený objekt představuje obsahovou část programu, dle návrhu jde o prostřední vrstvu.

Třída Main

V případě obsahu umístěného na ploše se většinou jedná o objekt třídy `MovieClip`, v případě této realizace je navrženo prostředí prostorové. Z toho důvodu třída `Main` dědí vlastnosti třídy `Sprite`, která je paměťově méně náročnou verzí `MovieClip`, pouze s jedním snímkem, ve kterém se odehrává veškeré řízení jeho obsahu.

Zobrazení obsahu je možné jen díky hlavním komponentám, které jsou součástí prostorového rozšíření `Paprrvision engine`:

- **scéna** typu `Scene3D`, do které jsou umístěny veškeré ostatní objekty prostřední vrstvy
- **prostor** typu `Viewport3D` představuje virtuální 3D prostor definovaného rozměru. Implementace využívá možnosti určit prostor automaticky podle rozměru zobrazovaného okna aplikace. Atribut `interactive` zajišťuje, že je prostor připraven k interaktivnímu použití.
- **kamera** typu `Camera3D` určuje místo v prostoru, ze kterého si pomyslně uživatel scénu prohlíží. Jde o neviditelný prostorový objekt se speciálními parametry, které jsou určující pro parametry zobrazení. Je např. možné nastavit sbíhavost perspektivy (šíří ohniska) od ortogonálního až po velmi široké zobrazení, což značně ovlivní optickou deformaci zobrazovaného objektu.
- **renderer** typu `BasicRenderEngine` zajišťuje výsledné zobrazení do plochy pozorované uživatelem.

Po inicializaci množství podpůrných objektů je možné vytvořit vlastní scénu pomocí funkce `init3D()`.

Funkce `init3D`

`Init3D()` představuje funkci, která vytváří a nastavuje prostorovou scénu. Mimo hlavních entit je zde také vytvořena plocha, na které je zobrazováno menu rozcestníku. Toto menu představuje materiál nanesený na povrch prostorové plochy. Materiál je průhledný, aby bylo možné vytvořit zakulacené rohy obdélníku. Interaktivita materiálu (atribut `interactive`) zajišťuje možnost ovládat obsah zobrazený na ploše. Zároveň je ploše přiřazen posluchač událostí prostorové scény zajišťující reakci na najetí nad plochu kurzorem myši.

Pomocí třídy `Tweener` je vytvořen nájezd plochy ze vzdáleného prostoru na její konečnou pozici. Po dokončení nájezdu je spuštěna animace materiálu představující nájezd menu rozcestníku.

Funkce `loop3D`

Změny prostorového obsahu se na výstupu projeví vždy, až když je předchozí snímek překreslen podle nové situace. Funkce `loop3D()` zajišťuje překreslení scény, je volána posluchačem události nového snímku, jež je nastaven na obnovovací frekvenci 25 fps (snímků za sekundu).

Dalším úkolem funkce je interaktivně reagovat na pozici kurzoru myši nad scénou tak, že je měněno natočení kamery v prostoru, atribut `rotationX`, `rotationY`, `rotationZ`. Mezi natočením kamery v současném a předchozím snímku je proveden plynulý nájezd, nastavený na délku trvání 0,7 sekundy, tím je zajištěn velice plynulý pohyb a zároveň dynamické natáčení plochy vůči uživateli.

Pro zajištění interakce tlačítek plochy je nutné každému zvlášť přiřadit posluchač události kurzoru myši pro nájezd nad tlačítko, sjezd z tlačítka a jeho zmáčknutí (aktivace).

Funkce `btn_release`

Řízení událostí, které se vykonají po aktivaci libovolného tlačítka plochy zajišťuje funkce `btn_release()`.

Tato funkce je volána z posluchače aktivace tlačítka, společně s parametrem události. Tak je možné rozlišit, o které tlačítko se jedná a zvolit akci, která mu přísluší. Tlačítka menu mají společnou vlastnost, že při jejich aktivaci je potřeba změnit obsah plochy nebo celé scény.

Pro změnu obsahu plochy, tj. materiálu jenž je na něm umístěn, bylo využito momentu, kdy je plocha od uživatele vzdálena a současně natočena kolmo. Proto při aktivaci tlačítka plocha odrotuje do prostoru a následně se vrací již s novým obsahem.

Jednotlivé plochy jsou tvořeny objektem typu `MovieClip`, do kterého jsou případně dynamicky nahrány potřebné informace dle konfiguračního souboru `objects.xml` vytvořeného podle

schématu pro načítání obsahu (navrženo výše). Načítáním obsahu podle konfiguračního souboru se zabývá speciální třída `myLoaderXML`, která bude blíže popsána v jedné z následujících kapitol.

Funkce `changeDesk`

Výběr měněného materiálu prostorové plochy probíhá pomocí funkce `changeDesk()`, která vybírá podle parametru příslušný konkrétní obsah.

Další důležitou úlohou, kterou tato funkce vykonává, je zajištění historie procházení prezentace. Z návrhu vyplývá, že struktura prezentace může být v některých částech dynamicky vytvářena na základě konfiguračního souboru. Současně má tlačítko zpět stále totožnou funkci, tedy vracet obsah o krok zpět. Tyto kroky musí být chronologicky zaznamenány, aby bylo možné se po nich případně vrátit. Za tímto účelem je vytvořeno pole `deskHistory`, které představuje zásobník hodnot určujících historii průchodu prezentací.

Právě funkce `changeDesk` zajišťuje vložení nového kroku do historie při změně obsahu. Zároveň, detekci stisku tlačítka zpět, je obsah vyměněn za materiál, který se vyskytuje na vrcholu zásobníku `deskHistory`. Při této změně je vrchol zásobníku vyjmut.

Nastavení materiálu provádí funkce `setNewMat()`, která je volána funkcí `changeDesk()`.

Funkce `modelChangeView`

Funkce `modelChangeView()` je vázána na prohlížení prostorového obsahu galerie. Zajišťuje interaktivní změny parametrů 3D objektů.

Detekuje případné stisknutí klávesy `Shift` na klávesnici, která má vnitřní kód 16, a následně upravuje velikost zobrazeného objektu. Konkrétní hodnota je vypočtena ze souřadnice kurzoru při stisku tlačítka myši a hodnoty rozměru výšky zobrazené plochy. Tato hodnota je přičtena k původnímu parametru objektu scény.

Detekce stisku klávesy `Ctrl`, s vnitřním kódem 17, způsobí interaktivní změnu pozice zobrazeného objektu. Změna je prováděna ve svislém i vodorovném směru podle počáteční souřadnice kurzoru myši při stisku jejího tlačítka.

V případě, že žádná klávesa stisknuta není, je, podobně jako předchozí parametry, upravována hodnota natočení objektu ve dvou osách X a Y.

Funkce `KeyUp`, `KeyDown`

Funkce `KeyUp()` a `KeyDown()` jsou téměř totožné, rozdílem je pouze druh události, na jejíž detekci příslušně reagují. Jde o zajištění možnosti výběru položek v menu rozcestníků klávesnicí. Funkce nejsou nutné pro celkový chod aplikace, drobným, ale o to příjemnějším způsobem rozšiřují uživatelský komfort v ovládání aplikace.

Téměř každá klávesa klávesnice (kromě F10) má interně ve Flashi přiřazen svůj kód, který je nutné znát při detekci aktivity příslušné klávesy.

Následující tabulka ukazuje přiřazení tlačítek zobrazených v menu rozcestníku k jejich odpovídajícímu kódu klávesy. Je patrné, že některé klávesy jsou sloučeny do jediného tlačítka, ale funkčně je reakce na jejich stisk shodná.

Tlačítko menu	Klávesa (běžné označení)	Tlačítko menu	Klávesa (běžné označení)
btn_0	57 (9)	btn_12	46 (Delete)
btn_1	48 (0)	btn_13	35 (End)
btn_2	187 (= +)	btn_14	75 (k)
btn_3	191 (/ ?)	btn_15	76 (l), 186 (; , ů) 188 (, ?), 190 (. :)
btn_4	8 (Backspace)	btn_16	77 (m)
btn_5	45 (Insert)	btn_17	189 (- _)
btn_6	36 (Home)	btn_18	16 (Shift)
btn_7	73 (i)	btn_19	38 (šipka nahoru)
btn_8	79 (o)	btn_20	32 (mezerník)
btn_9	80 (p)	btn_21	18 (pravý Alt)
btn_10	219 ([{ ú), 221 (] })) 222 (`` ' §), 220 (\ ')	btn_24	40 (šipka dolů)
btn_11	13 (Enter)		

Tabulka 6: Kódy kláves menu rozcestníku

Funkce makeModel

Funkce `makeModel()` zajišťuje nahrání a zobrazení příslušného modelu vybraného v galerii. Podle konfiguračního souboru je určeno, o jaký typ dat má jít, což vede ke spuštění obsluhy načítání a zobrazení.

- **typ 0 - Text**

Při nahrávání textu je vytvořeno textové pole obsahující data načtené položky.

- **typ 1 - Obráz**

O nahrání obrázku vybraného typu se stará objekt `ImageLoader` typu `Loader`. Posluchače událostí na něj napojené zajišťují obsluhu načítání dat z externího souboru dle adresy v konfiguračním XML.

- **typ 2 - Zvuk**

Je nahrán stejným způsobem jako obrázek, pouze je potřeba navíc zobrazit ovladač hlasitosti a možnost zvuk vypnout.

- **typ 3 - 3D**

Nahrání prostorových dat je prováděno podle schématu, které je uvedeno v příloze A.1 a A.2, v závislosti na verzi formátu. Kvůli potřebě zobrazit texturu je nutné, aby byla přiložena k modelu, případně byla zadána její adresa v konfiguračním XML.

- **typ 4 - Video**

Video je nahráno z externího souboru dle načtené adresy stejným způsobem jako zvuk. Ovládače jsou shodné s ovladači zvuku.

- **typ 5 - Ostatní**

Tento typ je určen především pro smíšené adresáře, ale při nahrávání jednotlivých souborů má také význam. Bez udání typu nahrávaných dat se aplikace pokusí zjistit formát dat dle jeho přípony. Jedná-li se o známý formát, je objekt nahrán, v opačném případě je uživatel na vzniklou skutečnost upozorněn.

4.6.3 Třída myLoaderXML

Tato třída je speciálně vytvořena pro potřebu načítání konfiguračního XML. Třída myLoaderXML dědí vlastnosti třídy MovieClip, což je výhodné především díky možnosti použití různých objektů pro načítání dat.

Hlavní smyčka načítá do objektu dataLoader konfigurační soubor. Posluchač událostí je nastaven tak, že při dokončení načítání je načtený objekt předán funkci dataNactena(). Následně je objekt parsován na typ XML nového objektu xmlData.

xmlData ve výsledku obsahuje strukturu načteného konfiguračního souboru a ostatních částí tak, že mohou ke každé jeho entitě či atributu přistupovat přes jméno elementu a jeho index.

4.6.4 Skript předzpracování 3D objektů

Skript není přímo spojen s výslednou aplikací, ale byl vytvořen pro potřebu převodu prostorového modelu do vhodného formátu zpracovávaného technologií Flash.

Na základě analýzy kompatibility byla pro zpracování skriptu zvolena verze Blender 2.5.2. Tato aplikace předurčuje nutnost vytvořit skript v jazyce Python verze 3.1. Pro starší verzi Blenderu je nutné skript upravit pro jazyk Python 2.6, ovšem úprava se musí dotknout i aplikačního rozhraní, které není napříč verzemi kompatibilní.

Práce s XML se opírá o modul expat, především jeho parser, který pro každý počáteční element XML volá funkci start_element, která načítá její atributy. Vnitřní obsahovou část elementů zpracovává funkce char_data, která načte její hodnotu.

Dále jsou již všechny potřebné údaje jednoho modelu načteny, proto je volána funkce convert, která provádí úpravu prostorového objektu. Funkce se opírá o aplikační rozhraní Blenderu skrze modul bpy. Tak je možné přistupovat k potřebným datům a prostředkům aplikace Blender.

Protože je skript určen pro postupné zpracování více modelů, je nejprve zajištěno, aby scéna neobsahovala žádné objekty, tedy případné objekty jsou smazány. Následně je do scény importován prostorový objekt formátu s příponou .obj. V této chvíli je nutné podotknout, že by bylo z hlediska této konkrétní prezentace výhodnější využít formátu VRML, ale Blender 2.5.2 zatím neobsahuje modul pro jeho načítání. Zároveň je formát OBJ také velice rozšířený a je podporován téměř všemi dostupnými 3D editory.

Prostorový objekt umístěný ve scéně má interně uložen rozměr ve všech třech osách. Tato vlastnost je přístupná jako atribut `dimensions` indexovaný 0 pro osu X, 1 pro Y a 2 pro Z. Postupů, jak určit výslednou velikost, může být více. Vzhledem k faktu, že prostorový objekt je v prezentaci počátečně natočen k uživateli plochou X,Y (rovnoběžně k rovině zobrazení), je zvolen jako určující rozměr právě průměr hodnoty `dimensions[0]` a `dimensions[1]`. Tak je zajištěno, že je v prezentaci objekt zobrazen ve vhodné velikosti.

Zvolená hodnota, v našem případě konkrétně 2, dělená předchozím průměrem čelních rozměrů, udává index z. Nový rozměr je výsledkem násobení původního rozměru a indexu z.

Výsledný objekt je nutné převést tak, aby byl jeho povrch reprezentován pouze trojúhelníky. Při použití exporteru do formátu COLLADA se však tento převod provádí automaticky, takže stačí pouze export provést.

Nový soubor, který je stejného jména jako vstupní, ale liší se příponou .dae, je uložen ve stejné složce jako soubor původní.

4.6.5 Video a animace

Jako součást prezentace bylo potřeba vytvořit i video, které tvoří obsahovou část. Mimo nasnímáných pohledů na prostorové objekty pomocí CamStudia, byla vytvořena prostorová animace v 3D modelovacím programu 3D's MAX, ve které se mají uplatnit některé zásady tvorby prezentace, konkrétně, aby byla daná část prezentace poutavá či zábavná. Detailní popis tvorby je za hranicí této práce, proto je zde uveden pouze stručně.

Prostorová animace je založena na tzv. klíčování pozic prostorových objektů a kamer. Scéna byla vytvořena pěti kamerami a vlastním modelem čítajícím 360 objektů představujících fiktivní družici letící kolem Země. Úkolem družice je nasměrovat uživatele na zeměpisnou polohu areálu FIT VUT v Brně, což činí vržením žluté vlaječky kybernetickou rukou směrem k Zemi.

Vytváření animace pomocí klíčů na určitých snímcích a jejich následné automatické propojení je způsob, který je téměř shodný pro tvorbu v ploše (Flash) i prostoru (3D's MAX). Rozdíl při generování výstupu je však velký. Flash je celkově vystavěn na principu snímků, animace se tak vytváří po snímcích za běhu aplikace. 3D animace s velkým množstvím detailů a nastavením složitých materiálů je nutné renderovat delší dobu. Zde konkrétně trvalo renderování animace cca 40 hodin, proto byla rozdělena do více jednotlivých vrstev, které se ve výsledném videu spojily.

5 Diskuse výsledku

Výsledkem praktické části diplomové práce je aplikace, která demonstruje možnosti prezentace s užitím videa a prostorových objektů.

Mimo hlavní aplikaci však bylo vytvořeno několik podpůrných prostředků. Je to především skript pro předzpracování 3D objektů a jejich převod do formátu použitelného ve vytvořené aplikaci, dále modul pro vkládání komentářů do videa. Součástí prací bylo také testování použitelnosti a dalších vlastností 3D scanneru.

Testování 3D scanneru bylo prováděno v několika dnech (celkově cca 15 hodin), přičemž bylo provedeno přibližně 100 jednotlivých scanů. Hodnoty není možné uvést přesně, protože, převážně v počátku testování, nebyla ukládána všechna data, ale pouze referenční, tak aby bylo možné např. sledovat případné zlepšení kvality výstupu při změně nastavení parametrů snímání. Výsledky a doporučení jsou uvedeny v kapitole 3.2. Ukázky výstupů 3D scanneru jsou součástí obrazové přílohy B.1.

Vytvořený skript v Pythonu je možné použít pro automatický převod formátů a předzpracování modelů pro použití v prezentaci. Skutečností však je, že tento postup je výrazněji složitější než použití formátu VRML a jeho zobrazení pomocí specializovaných prohlížečů, které jsou dostupné i bezplatně. Proto lze použití technologie Flash pro prezentaci prostorových dat doporučit spíše u aplikací zaměřených na vysokou interaktivitu, ale se stálými modely (vstupní data se příliš nemění) např. hry nebo prostorové uživatelské rozhraní.

Z výsledku je patrné, že je možné pro tvorbu interaktivních multimediálních prezentací použít technologii Flash, ovšem spojení více typů médií je poměrně problematické. Vzniklý projekt většího rozsahu se může stát z pohledu vývoje nepřehledným, a to i s využitím vývojového prostředí Flex. Skutečnost, že je třeba všechna média opatřit zvláštním uživatelským rozhraním, k jednoduchosti produkce případné prezentace také nepřispívá. Další problémy mohou vznikat při skládání složitějších modulů, kdy je třeba věnovat zvýšenou pozornost uvolňování paměti. Propojení interaktivních materiálů s prostorovými prvky také není zcela intuitivní a je potřeba dodržovat strukturu zanoření prvků na co nejjednodušší úrovni.

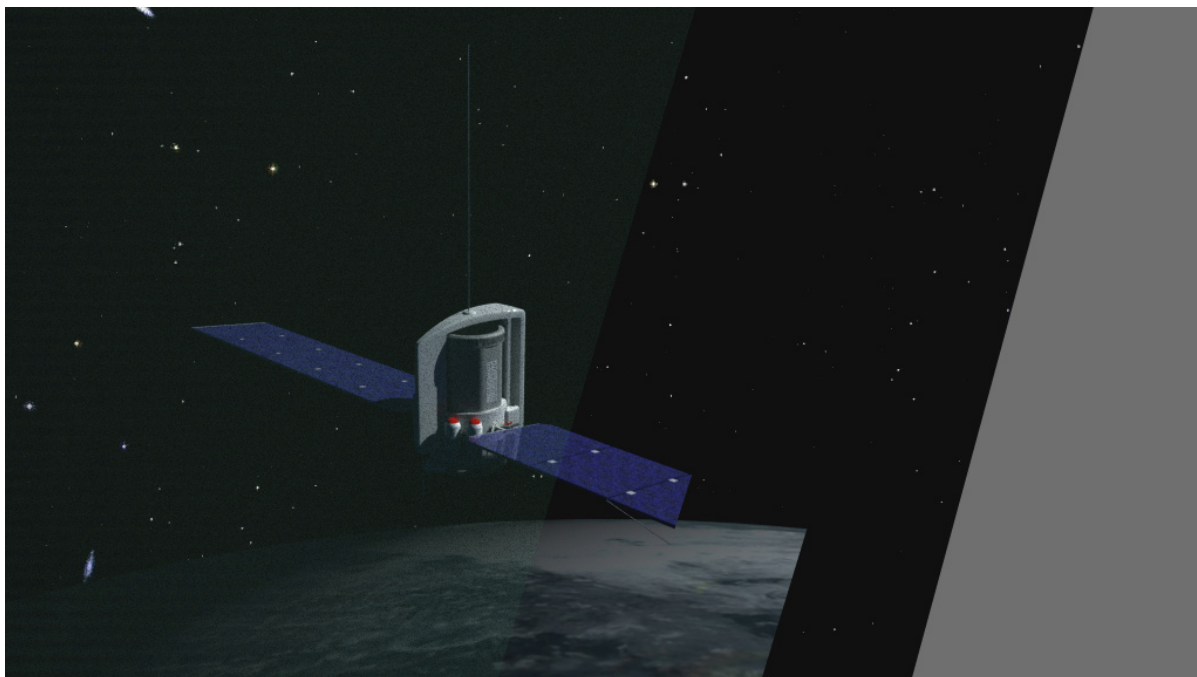
Výhodou však zůstává možnost tvorby jednodušších stavebních modulů, které je možné využít i samostatně či jako komponenty různých projektů. Takovým modulem je vytvořený `load_VS.swf`, který slouží k nahrávání příslušného videa, zároveň ale nahrává soubor XML s popisky.

Implementovaná vlastnost zobrazení textu ve videu je alternativní možností k použití titulků. Zde jsou však vlastnosti rozšířeny o možnost umístit titulek na libovolnou pozici ve snímku, upravit jeho formátování pomocí HTML (např. barvu, velikost, apod.) a vložit do jednoho snímku libovolný počet titulků. Těchto vlastností se dá použít např. k jednoduchému komentování děje ve videu.

Ve výsledné aplikaci se podařilo skloubit využití textu, videa, 3D objektů, obrázků, animací a plochého i prostorového uživatelského rozhraní. Její podoba je částečně vidět na následujících obrázcích.



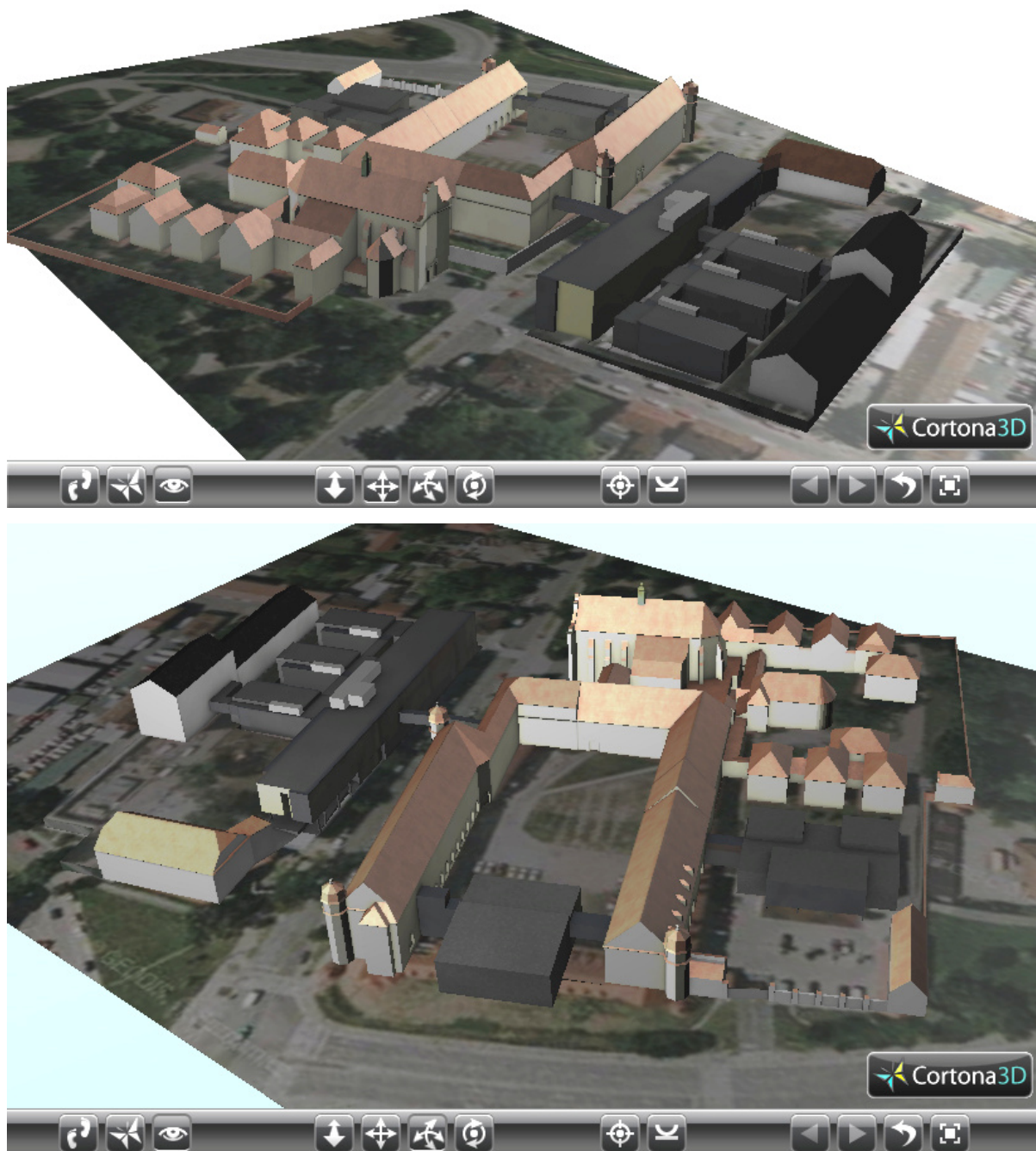
Obrázek 5.1: Ukázky vzhedu hlavního menu aplikace. Plocha, na které je menu umístěno se natáčí v prostoru za kurzorem. Na spodním obrázku je zobrazeno barevné ladění zvýraznění tlačítek při nájezdu myši. Reálně je však zvýrazněno pouze jedno tlačítko.



Obrázek 5.2: Jeden ze snímků prostorové animace, která je součástí vytvořeného videa. V obrázku jsou naznačeny vrstvy, ze kterých se výsledné video skládá.



Obrázek 5.3: Snímek z výsledného videa - obrazovka družice



Obrázek 5.4: Pohled na Areál FIT VUT v Brně. Původní prostorový model Ing. Radka Bartoně byl v aplikaci Blender převeden do formátu VRML a zobrazen prohlížečem Cortona 3D.

6 Závěr

Cílem práce bylo zhodnotit stav technologií použitelných pro tvorbu prezentace s pokročilými prvky a demonstrovat možnosti jejich použití pro prezentaci videa, prostorových scén a objektů.

Při studiu dodkladů bylo zjištěno, že na problematiku je nutné nahlížet jako na soubor témat několika oborů. Komplexním řešením se v současnosti velký počet zdrojů nezabývá, spíše je vždy řešena otázka jednotlivých podproblémů, které je možné s využitím vhodných nástrojů spojit do většího celku. Tímto směrem se také ubírala teoretická a následně i praktická část práce.

Byly zjištěny určité nedostatky použitelných prostředků, zejména v otázce kompatibility dílčích částí, které na sebe navazují v procesu tvorby prezentace, a také problém s načítáním dat při použití platformy Flash. Z toho důvodu byl navržen skript, který provádí konverzi formátů prostorových dat tak, aby byla použitelná s technologií Flash, tvořící základ hlavní části realizace. Současně byly také rozšířeny možnosti prezentace načítat dynamicky multimediální obsah bez nutnosti kompilace a vkládání textových popisků do videa.

Prezentace byla vytvořena za účelem demonstrace funkčního řešení spojujícího větší počet multimediálních prvků na téma propagace Ústavu počítačové grafiky a multimédií.

Vytvoření prezentace s sebou neslo i nutnost vytvořit mimo prezentačního programu i samotný prezentovaný obsah. V tomto ohledu bylo vhodné vyzkoušet tvorbu prostorových modelů pomocí 3D scanneru, který byl v dané době zcela novým zařízením, takže jeho použití představovalo i jeden z jeho prvních testů. Doporučení, která z testů vyplynula, jsou také součástí této zprávy. Součástí videí obsažených v prezentaci je i vizualizace vlastní prostorové animace a tvorba videí z prostorových dat pro prezentace s omezenými možnostmi interakce uživatele.

Výsledná aplikace je použitelná jako desktopová aplikace spouštěná na operačním systému Windows, dále jako součást webové stránky a v neposlední řadě je vytvořena i verze pro distribuci na DVD. Podařilo se tak vytvořit řešení splňující původní záměr komplexní prezentace s pokročilými prvky.

S ohledem do budoucna by bylo přínosné analyzovat sledovanost a zájem uživatelů. Aplikaci je možné rozšířit o další moduly, jejichž funkce by znamenaly přetvoření prezentace na dynamickou webovou stránku či přímo webový portál. Stejně tak je možné použít některý ze stávajících modulů pro tvorbu aplikace zaměřené jiným směrem. Spojení formátu XML s možností jeho automatického generování např. webovými servery, nabízí využití modulu generování textu ve videu pro tvorbu výukových a informačních klipů.

Literatura

- [1] Rapoza, J.: Ria war is brewing [online]. 2008. Dostupné na URL: http://etech.eweek.com/content/application_development/ria_war_is_brewing.html (květen 2010)
- [2] Cover, R.: VRML (Virtual Reality Modeling Language) and X3D [online]. 2006. Dostupné na URL: <http://xml.coverpages.org/vrml-X3D.html> (květen 2010)
- [3] Tišnovský, P.: XML + 3D = X3D [online]. 2008-01-24. Dostupné na URL: <http://www.root.cz/clanky/xml-3d-x3d/> (květen 2010)
- [4] Borgione, J.: Free streaming video software [online]. 2007. Dostupné na URL: <http://camstudio.org/> (květen 2010)
- [5] Vávru, V.: Pár střípků o Silverlight [online]. 2007-09-30. Dostupné na URL: <http://vavru.cz/ostatni/par-stripku-o-silverlight/> (květen 2010)
- [6] Kačmář, D.: Novinky v Silverlight 3.0 - finální verze je venku [online]. 2009-07-14. Dostupné na URL: <http://blogs.msdn.com/vyvojari/archive/2009/07/14/novinky-v-silverlight-3-0-fin-ln-verze-je-venku.aspx> (květen 2010)
- [7] Špulák, O.: Microsoft vydal beta verzi Silverlight 4 [online]. 2009-12-03. Dostupné na URL: <http://www.slunecnice.cz/tipy/microsoft-vydal-beta-verzi-silverlight-4/> (květen 2010)
- [8] Waldron, R.: The Flash history: How it began [online]. 2000-11-20. Dostupné na URL: http://www.flashmagazine.com/news/detail/the_flash_history/ (květen 2010)
- [9] Kolektiv autorů: Collada - Interchange Format Release 1.1.0 [online]. Tokyo, 2004. Dostupné na URL: http://www.collada.org/public_forum/files/COLLADA_Specification_110.pdf (květen 2010)
- [10] Gaudin, S.: Intel guru says 3-D Internet will arrive within five years [online]. 2010-04-08. Dostupné na URL: <http://www.computerworld.com/s/article/9175048/> (květen 2010)
- [11] Kolektiv autorů: Top 15 Most Popular Web 2.0 Websites [online]. 2010-05. Dostupné na URL: <http://www.ebizmba.com/articles/web-2.0-websites> (květen 2010)
- [12] Ruel, L. et al.: Recall of Information Presented in Text vs. Multimedia Format [online]. 2000. Dostupné na URL: <http://www.poynterextra.org/EYETRACK2004/multimediarerecall.htm> (květen 2010)

- [13] Dutch, D.: Businesses Embracing Online Video Will Fuel The Web's Revolution [online]. 2009-01-21. Dostupné na URL: http://www.mediapost.com/publications/?fa=Articles.showArticle&art_aid=98807 (květen 2010)
- [14] Graham, J.: Twelve ways to make a better presentation than anyone else, American Salesman [online]. 2002-03-01. Dostupné na URL: <http://www.allbusiness.com/sales/sales-presentations/125554-1.html> (květen 2010)
- [15] Chapman, A.: Presentations skills [online]. 2010. Dostupné na URL: <http://www.businessballs.com/presentation.htm> (květen 2010)
- [16] Long, B. et al.: Velká kniha digitálního videa. Praha: Computer Press, 2005. ISBN 80-251-0580-6
- [17] Kolektiv autorů: Vivid 910 [online]. Osaka, 2002. Dostupné na URL: http://www.konicaminolta.com/content/download/1104/7683/vivid910_e9.pdf (květen 2010)
- [18] Gargulák, D.: Animace algoritmů v prostředí Silverlight, diplomová práce, Brno, FIT VUT v Brně, 2009, s. 6-9.
- [19] Biron, P. et al.: XML Schema Part 2: Datatypes. W3C, 2004. Dostupné na URL: <http://www.w3.org/TR/xmlschema-2/> (květen 2010)
- [20] Bray, T. et al.: Extensible Markup Language (XML) 1.0 (Second Edition). W3C, 2008. Dostupné na URL: <http://www.w3.org/TR/REC-xml/> (květen 2010)
- [21] Bray, T. et al.: Namespaces in XML. W3C, 2009. Dostupné na URL: <http://www.w3.org/TR/REC-xml-names/> (květen 2010)

Seznam příloh

Příloha A

- A.1 Zobrazení Collada 1.4.0
- A.2 Zobrazení Collada 1.4.1
- A.3 XML Schema (WXS)
- A.4 Příklad použití 3D scanneru v praxi

Příloha B

- B.1 Výstupy 3D scanneru

Příloha - CD

- Programová dokumentace
- Zdrojové kódy všech částí
- Spustitelná aplikace

Příloha A

A.1 Zobrazení Collada 1.4.0

```
package {
    import flash.events.Event;
    import org.papervision3d.lights.PointLight3D;
    import org.papervision3d.materials.shadematerials.GouraudMaterial;
    import org.papervision3d.materials.utils.MaterialsList;
    import org.papervision3d.objects.parsers.DAE;
    import org.papervision3d.view.BasicView;

    public class DaeParser_v1 extends BasicView {
        private var _dae:DAE;
        private var _light:PointLight3D;

        public function DaeParser_v1() {
            lights();
            init3d();
        }

        private function lights():void {
            // vytvoření objektu světla
            _light = new PointLight3D(true, true);
            // umístění světla do pozice kamery
            _light.copyPosition(camera);
        }

        private function init3d():void {
            // vytvoření seznamu materiálů
            var materialsList:MaterialsList = new MaterialsList();
            // použití materiálu s Gouraudovým stínováním
            var material:GouraudMaterial = new GouraudMaterial(_light);
            materialsList.addMaterial(material, "all");
            // inicializace a načtení 3D obsahu
            _dae = new DAE();
            _dae.load("objekt.dae", materialsList);
            // přidání objektu do scény
            scene.addChild(_dae);
            startRendering();
        }

        override protected function onRenderTick(event:Event = null):void {
            super.onRenderTick();
        }
    }
}
```

A.2 Zobrazení Collada 1.4.1

```
package {
    import flash.events.Event;

    import org.papervision3d.lights.PointLight3D;
    import org.papervision3d.materials.shadematerials.GouraudMaterial;
    import org.papervision3d.materials.utils.MaterialsList;
    import org.papervision3d.objects.parsers.Collada;
    import org.papervision3d.view.BasicView;

    public class ColladaParser_v1 extends BasicView {
        // objekt scény
        private var _collada:Collada;
        private var _light:PointLight3D;

        public function ColladaParser_v1() {
            lights();
            init3d();
        }

        private function lights():void {
            // vytvoření objektu světla
            _light = new PointLight3D(true, true);
            // umístění světla do pozice kamery
            _light.copyPosition(camera);
        }

        private function init3d():void {
            // vytvoření seznamu materiálů
            var materialsList:MaterialsList = new MaterialsList();
            // použití materiálu s Gouraudovým stínováním
            var material:GouraudMaterial = new GouraudMaterial(_light);
            materialsList.addMaterial(material, "all");
            // inicializace a načtení 3D obsahu
            _collada = new Collada("objekt.DAE", materialsList, 0.05);
            // přidání objektu do scény
            scene.addChild(_collada);
            startRendering();
        }

        override protected function onRenderTick(event:Event = null):void {
            super.onRenderTick();
        }
    }
}
```

A.3 XML Schema (WXS)

A.3.1 Načítání obsahu

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!-- soubor -->
  <xsd:complexType name="item">
    <xsd:sequence maxOccurs="1" minOccurs="0">
      <xsd:element name="textData" minOccurs="0" maxOccurs="1"
type="xsd:string"/>
    </xsd:sequence>
    <xsd:attribute name="jmeno" use="required" type="xsd:string"/>
    <xsd:attribute name="typ" use="required" type="xsd:unsignedInt"/>
    <xsd:attribute name="version" use="optional" type="xsd:string"/>
    <xsd:attribute name="addr" use="optional" type="xsd:string"/>
  </xsd:complexType>
  <!-- trida s objekty -->
  <xsd:group name="class">
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:element name="folder" minOccurs="0" maxOccurs="unbounded"
type="category"/>
      <xsd:element name="file" type="item"/>
    </xsd:choice>
  </xsd:group>
  <!-- korenovy objekt-->
  <xsd:complexType name="root">
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:group ref="class"/>
    </xsd:choice>
  </xsd:complexType>
  <!-- kategorie-->
  <xsd:complexType name="category">
    <xsd:choice maxOccurs="unbounded" minOccurs="0">
      <xsd:group ref="class"/>
    </xsd:choice>
    <xsd:attribute name="jmeno" type="xsd:string"/>
  </xsd:complexType>
  <xsd:element name="objects" type="root"/>
</xsd:schema>
```

A.3.2 Generování textu ve videu

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!-- polozka -->
  <xsd:complexType name="item_c">
    <xsd:sequence maxOccurs="1" minOccurs="0">
      <xsd:element name="textData" minOccurs="0"
maxOccurs="unbounded" type="xsd:string"/>
    </xsd:sequence>
    <xsd:attribute name="xpos" use="optional" type="xsd:unsignedInt"/>
    <xsd:attribute name="ypos" use="optional" type="xsd:unsignedInt"/>
  </xsd:complexType>
  <!-- snimek -->
  <xsd:complexType name="frame_c">
```

```

        <xsd:sequence>
            <xsd:element name="item" type="item_c" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="number" type="xsd:unsignedInt"
use="required"/>
    </xsd:complexType>
    <!-- korenovy objekt-->
    <xsd:complexType name="root">
        <xsd:sequence>
            <xsd:element name="frame" type="frame_c" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- root-->
    <xsd:element name="titles" type="root"/>
</xsd:schema>

```

A.3.3 Skript předzpracování

```

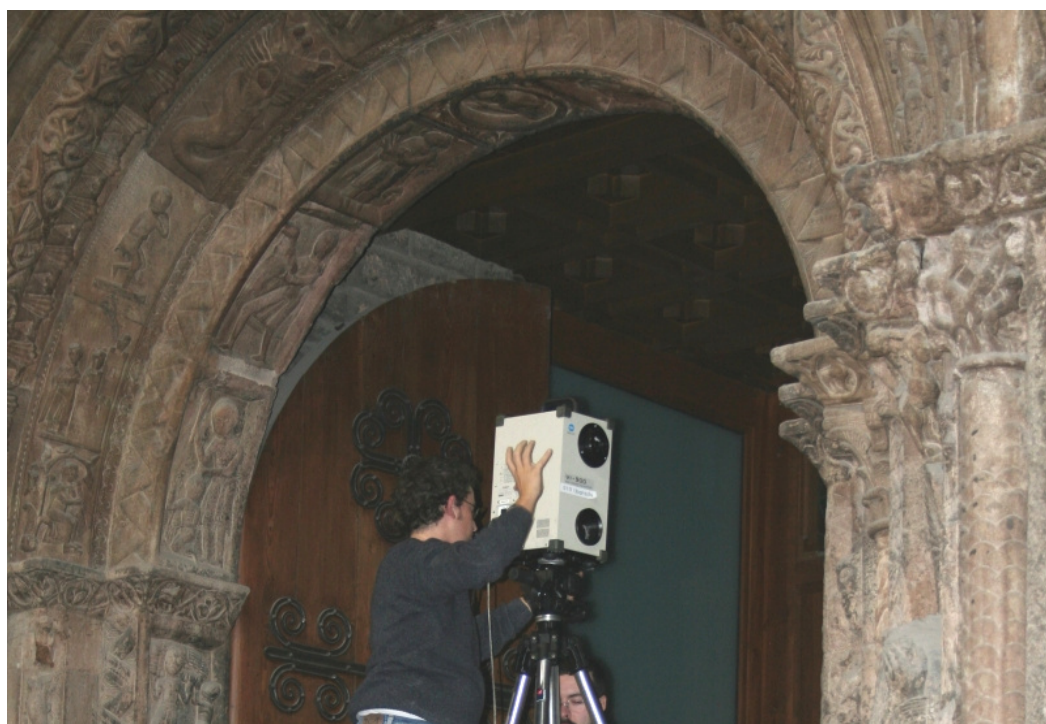
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <!-- trida s objekty -->
    <xsd:complexType name="object_c">
        <xsd:sequence>
            <xsd:element name="textData" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="jmeno" type="xsd:string" use="required"/>
    </xsd:complexType>
    <!-- korenovy objekt-->
    <xsd:complexType name="root">
        <xsd:sequence>
            <xsd:element name="object" type="object_c" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- objekt pro adresar-->
    <xsd:element name="models" type="root"/>
</xsd:schema>

```


A.4 Příklad použití 3D scanneru v praxi



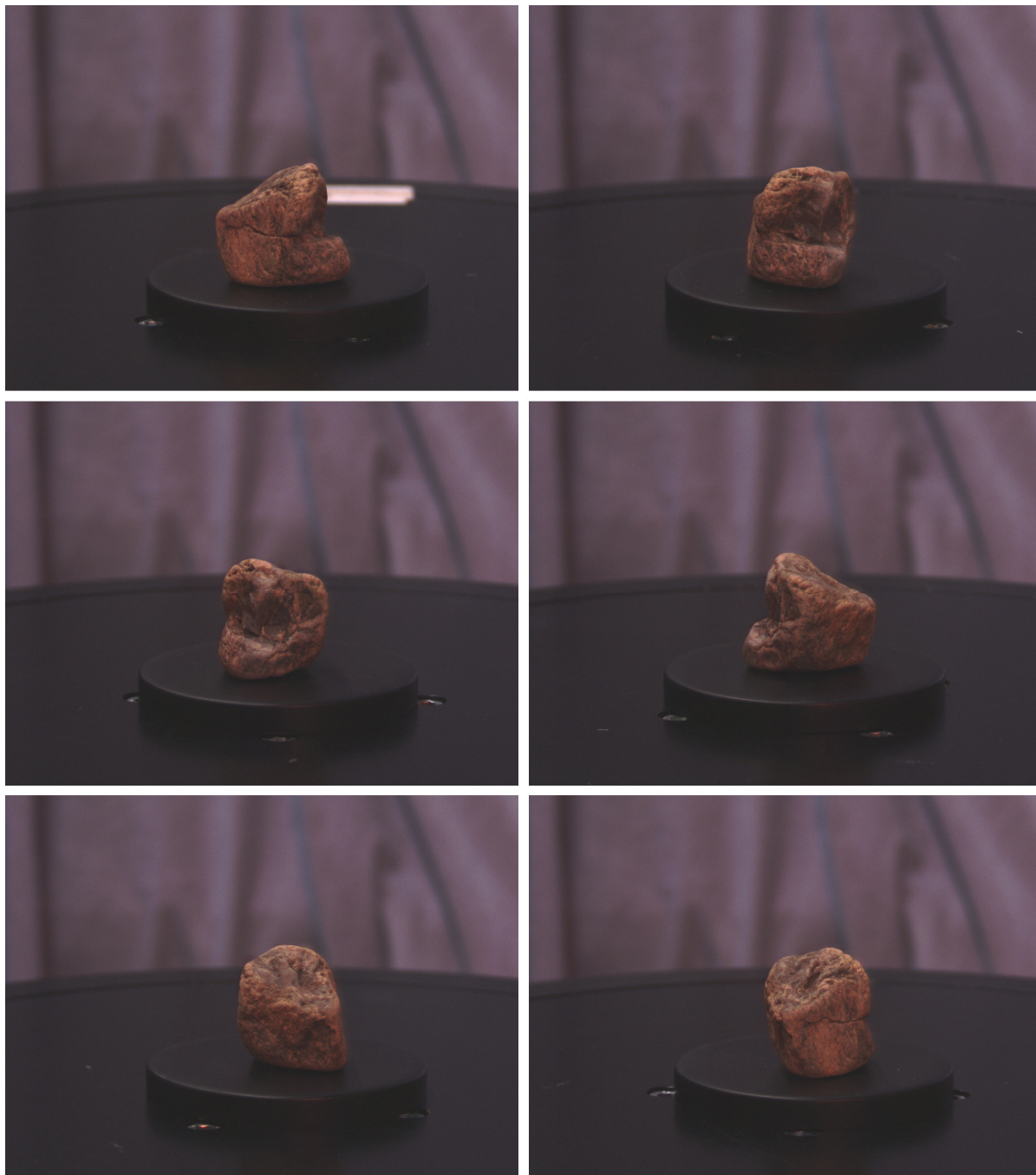
Obrázek 1: Ukázka vhodného prostředí pro 3D skenování
(Hampson Archeological Museum State Park)



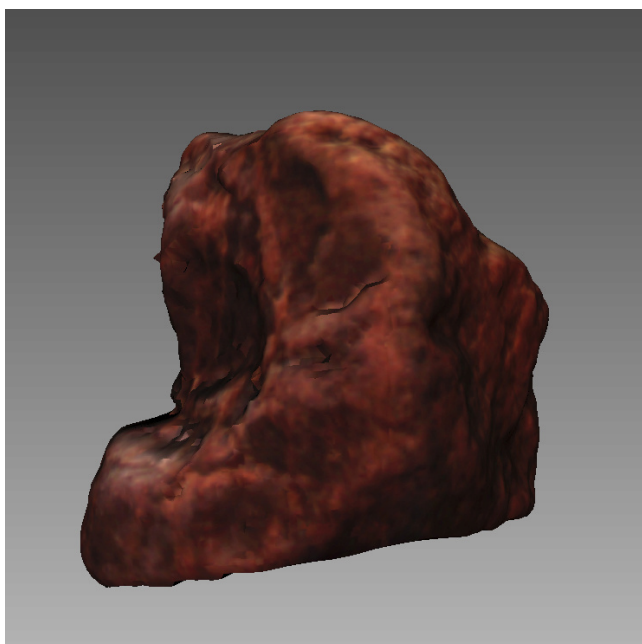
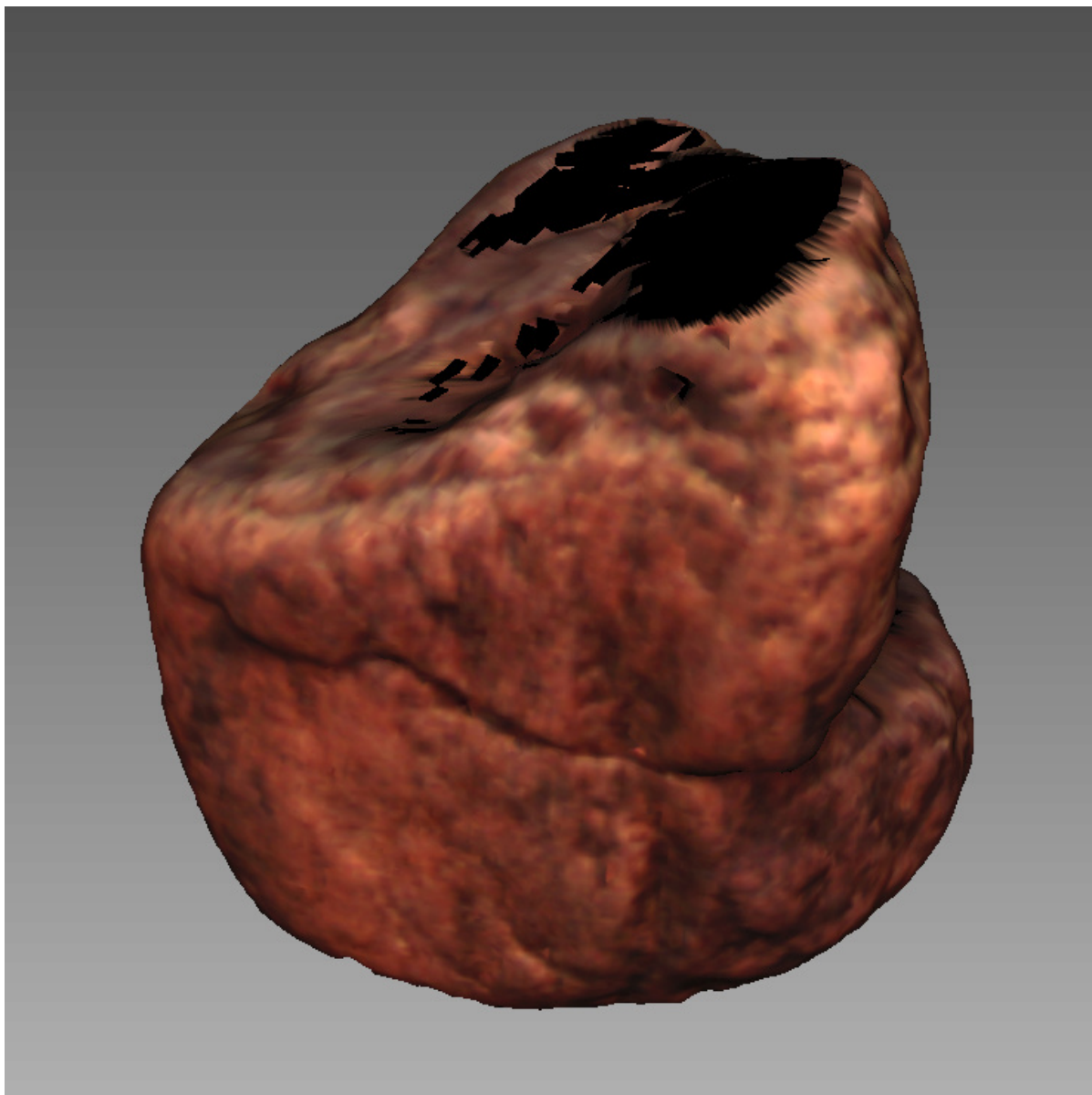
Obrázek 2: Použití 3D scanneru v exteriéru
(Matteo Dellepiane Visual Computing Lab, ISTI-CNR Pisa)

Příloha B

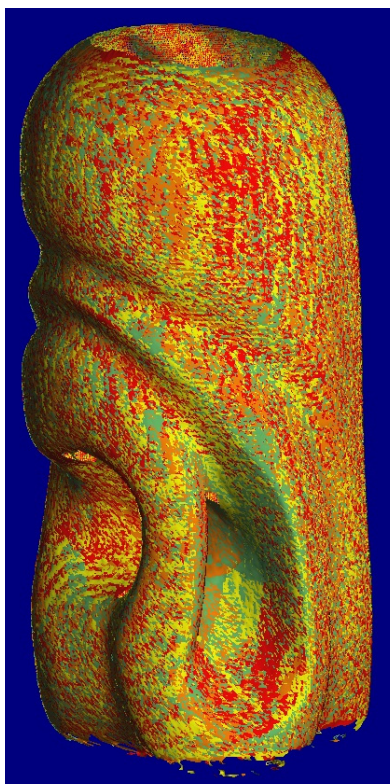
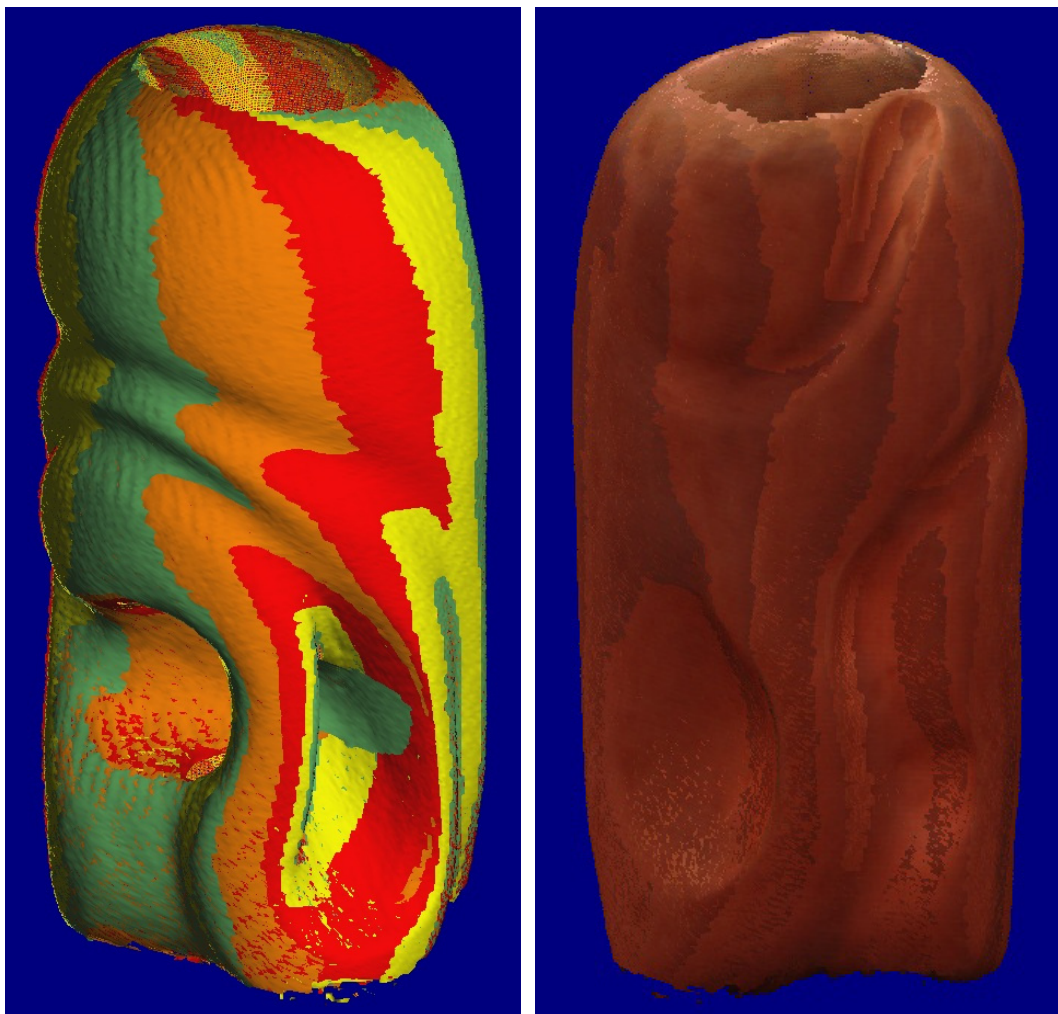
B.1 Výstupy 3D scanneru



Obrázek 3: Fotografie kamene (5 x 5 x 4,5 cm) pořízené 3D scannerem pro potřeby tvorby textury. Kancelářské osvětlení, automatické ostření a odhad vzdálenosti zapnuto. Automatická rotace, krok 60°.

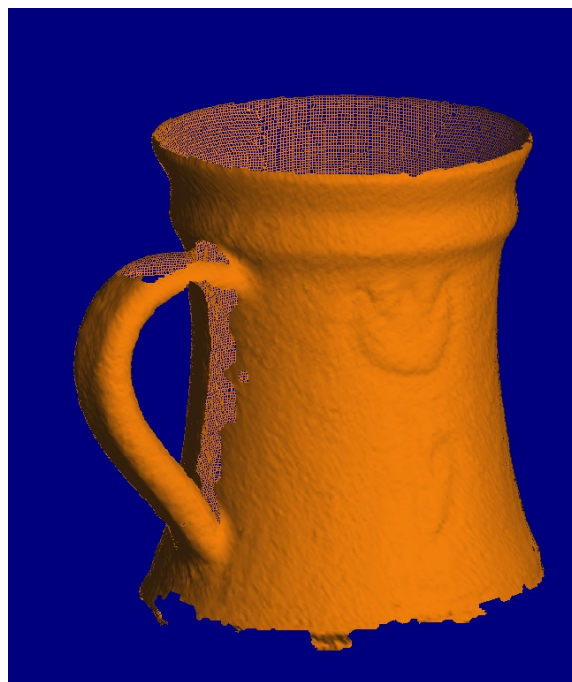
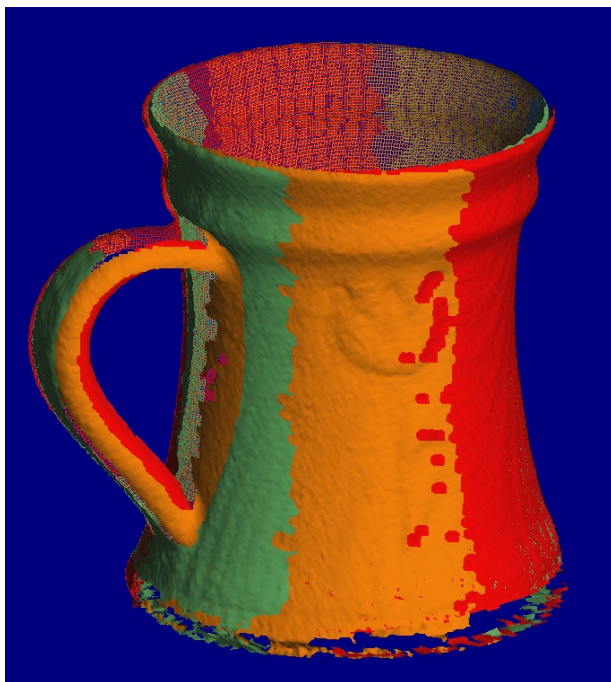


Obrázek 4: Zobrazení VRML (Cortona 3D). Nahoře je vidět problém snímání povrchu laserem. V místě díry v povrchu je nanесena černá textura i v případě, že původní textura na jejím místě data obsahuje.



Obrázek 5: Ukázky skenu dřevěného objektu (16,5 x 7 cm) v prostředí Polygon Editing Tool. Nahoře vlevo je vidět vyosení jednotlivých dílů při otáčení modelu. Nahoře vpravo objekt s nanesenou texturou. Na obrázku vlevo jsou části prostorově sesouhlaseny.

Na objektu je patrný problém v horní části, kde si automatické vyplňování děr nedokáže s takto komplikovaným tvarem poradit.



Obrázek 6: Keramický hrnek (9,5 x 7,5 cm)

Problém snímání předmětů tvaru otevřeného válce je jejich vnitřní stěna. Dalším problémem jsou složitější tvary tvořící v některé ose převis či most.

Na tomto modelu je patrný problém s lesklým glazovaným povrchem, který podél spodní hrany tvoří 0,5 cm široký pásek.